HAL
open science

Design of mechatronic products based on user-customized configuration : an application for industrial robots

Jing Li

To cite this version:

Jing Li. Design of mechatronic products based on user-customized configuration : an application for industrial robots. Mechanics [physics.med-ph]. Université de Technologie de Compiègne; Northwestern Polytechnical University (Chine), 2015. English. NNT : 2015COMP2211 . tel-01473955

HAL Id: tel-01473955

https://theses.hal.science/tel-01473955v1

Submitted on 22 Feb 2017

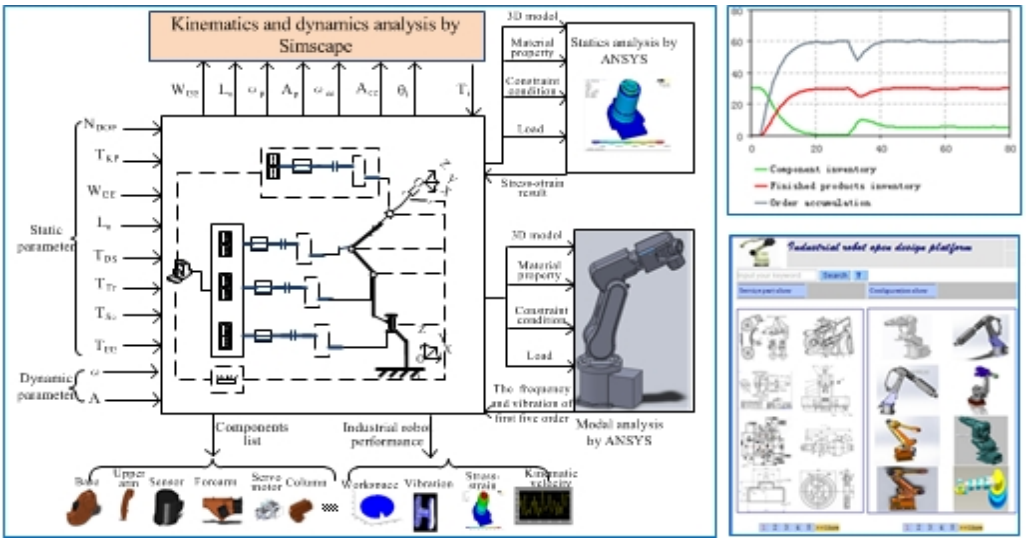
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Par **Jing LI**

Design of mechatronic products based on user-customized configuration : an application for industrial robots

Thèse présentée en cotutelle pour l'obtention du grade de Docteur de l'UTC



Soutenue le 28 août 2015
Spécialité : Mécanique Avancée

D2211

THÈSE

Pour l'obtention du grade de docteur délivré par

SORBONNE UNIVERSITÉS - UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

SPÉCIALITÉ MECANIQUE AVANCÉE

Ecole doctorale n°71 Sciences pour l'ingénieur - Laboratoire Roberval (CNRS, UMR 7337)

Présentée et soutenue publiquement par

Jing LI

Le 28 août 2015

Design of Mechatronic Products Based on User-Customized Configuration: An Application for Industrial Robots

DIRECTEURS DE THÈSE :

M. Benoît EYNARD

Mme Shurong Tong

JURY

M. BUDINGER Marc, Maître de Conférences - HDR, INSA de Toulouse	rapporteur
M. HE Zhen, Professeur, Tianjin University	rapporteur
M. EYNARD Benoît, Professeur des Universités, Université de Technologie de Compiègne	directeur de thèse
Mme TONG Shurong, Professor, Northwestern Polytechnical University	co-directrice de thèse
Mme PRELLE Christine, Professeur des Universités, Université de Technologie de Compiègne	examinatrice
M. QIN Xiansheng, Professor, Northwestern Polytechnical University	examinateur
Mme BOSCH-MAUCHAND Magali, Maître de Conférences, Université de Technologie de Compiègne	examinatrice
Mme DAABOUL Joanna, Maître de Conférences, Université de Technologie de Compiègne	examinatrice

ACKNOWLEDGEMENTS

First and foremost I would like to thank my supervisors Professor Benoît EYNARD in France and Professor Shurong TONG in China, who set this research program in motion and has offered timely guidance and support throughout its duration. Merci beaucoup!

Special thanks Professor Xiansheng Qin. In the process of my thesis writing, he gave me a lot of help, so that I can quickly determine the research direction. When I appeared puzzled, Professor Qin always found strategically advantageous position, and sharply pointed out where the problem is. Every time when I ask questions to Professor, I have been impressed by his profound knowledge.

Thanks Joanna Daaboul, Magali Bosch-Mauchand, Nadège Troussier, Keqin Wang, Xinwei Zhang, Matthieu Bricogne. They often proposed valuable advice to my research work. Thanks senior brother Zhaohua Wang, senior sister Chunzhi Li. Their diligence and prudence always remind me to keep trying. Thanks office colleagues Jun Du, Hongye He, Shan Yan, Jianfeng Jiang, Zongxiao Wu, Tianhang Liu, Jing Luo, Yanfen Cao, Deteng Liu. When I encounter difficulties, they always give me help.

Last, but certainly not least, thanks my parents, my brother, and my husband. They always give me confidence and encouragement. Thanks my daughter Kiki. She adds a lot of bright for my doctoral life, so that I can have a happy mood every day.

Sincerely thank the reviewers!

DESIGN OF MECHATRONIC PRODUCTS BASED ON USER-CUSTOMIZED CONFIGURATION: AN APPLICATION FOR INDUSTRIAL ROBOTS

Jing LI

ABSTRACT

In today's market, obtaining a variety of products through configuration design has become increasingly common. However, with the development of the market, customers have not only satisfied the company offering a variety of products, and more demands on participating in the process of configuration design by themselves, so that they can obtain fully personalized products. Customer participation leads to the changes of design process, company's management model, etc. Based on the above problem, this thesis takes industrial robot as an example, and studies the management issues related to the customer involved in the design, in order to address the contradiction between product diversification, personalized requirements and the long design cycle and high manufacturing costs.

Firstly, Auser-customized configuration design pattern is presented. The theory source of user-customized configuration design pattern is introduced, and then the related concepts are expounded. The corresponding business mode of user-customized configuration design pattern is given, and the key technologies to realize business mode is studied. System dynamics models were established for user-customized configuration design business mode and for traditional business mode of industrial robots by Anylogic simulation software.

Secondly, the component-based theory and method are studied, including the formal description of things, ontology representation, componentization and servitization. On this basis, the componentization description model is established for the product parts. And the model is represented as service-component. Next, the formation process and extension method of service-component are introduced. An example of industrial robot components modeling is analyzed, including establishing industrial robot domain ontology by protégé, describing, instantiating and extensing components.

Thirdly, the industrial robot user-customized configuration design template is constructed,

and users can obtain the industrial robot meeting constraints through parameters setting; The kinematics and dynamics analysis on template is taken by Simscape model, and the dynamic parameters is analyzed, and the finite element analysis on template is taken by ANSYS, including statics analysis and modal analysis. The parameters flow process in template is analyzed. Then taking industrial robot user-customized configuration design using configuration template as an example, the configuration template is analyzed in application.

Fourthly, the internal algorithm of user-customized configuration design is researched. Platform-based and user-leading user-customized configuration design process is constructed, and then the internal algorithm to keep the design running smoothly is studied, including the degree of freedom determination, fuzzy demand calculation, and service-component configuration and the configuration program evaluation. A case analysis is also taken for the internal algorithm

Finally, on the basis of the previous section, the prototype system design of the open design platform is taken. Based on system requirements analysis and system design, the main pages of the platform are designed, and the key functions are introduced

Keywords: configuration design; design management; user-customized design; industrial robot; ontology; object-oriented representation; product template

TABLE OF CONTENTS

Page

CHAPTER 1 INTRODUCTION 1
1.1 Background 1
1.2 The research status and analysis 2
1.2.1 The research status 2
1.2.2 The research status summary and analysis 15
1.3 Thesis significance and main research contents 16
1.3.1 Research significance 16
1.3.2 The structure of research contents 17
CHAPTER 2 USER-CUSTOMIZED CONFIGURATION DESIGN PATTERN 20
2.1 Theoretical Basis 20
2.2 Related Concepts of User-Customized Configuration Design Pattern 22
2.3 Business Mode Analysis 28
2.3.1 User-customized configuration design business mode 28
2.3.2 Key technologies of user-customized configuration design business mode 30
2.4 Validation of Proposed User-Customized Configuration Design Pattern through Simulation 32
2.4.1 Analysis of traditional business mode and user-customized configuration design business mode 32
2.4.2 System dynamics model 34
2.4.3 Simulation results and analysis 36
2.5 Chapter Summary 40
CHAPTER 3 COMPONENT-BASED MODELING FOR PRODUCT PARTS 42
3.1 Component-based theory and method 42
3.1.1 Formal description method of things 42
3.1.2 Ontology and domain of discussion 43
3.1.3 Componentization and servitization 44
3.2 Component-based modeling for product parts 45
3.2.1 The object model of component 45

3.2.2 The business model of component 46
3.3 The servitization of product component 46
3.3.1 Representation based on ontology 46
3.3.2 Description method for service-component 47
3.3.3 The service-component instantiation 48
3.4 The formation and extension of service-component 49
3.4.1 The formation process of service-component 49
3.4.2 The extension mechanism of service-component 50
3.5 Case Study: Industrial robot ontology and core components modeling 53
3.5.1 Industrial robot ontology modeling 53
3.5.2 Service-component servitization for industrial robot 56
3.6 Chapter Summary 60
CHAPTER 4 USER-CUSTOMIZED CONFIGURATION DESIGN TEMPLATE FOR INDUSTRIAL ROBOT 61
4.1 Industrial Robot Composition Structure 61
4.2 User-customized configuration design template establishment 63
4.2.1 Template graphic elements 63
4.2.2 Template Parameters 65
4.2.3 Internal constraint of template 66
4.3 Kinematics and dynamics analysis of template parameters 67
4.4 Finite Element Analysis of template Parameters 70
4.4.1 Statics Analysis 71
4.4.2 Modal analysis 72
4.5 Template operating principle 73
4.6 Template analysis example 74
4.7 Chapter Summary 79
CHAPTER 5 INTERNAL ALGORITHM FOR USER-CUSTOMIZED CONFIGURATION DESIGN 80
5.1 user-customized configuration design process analysis 80
5.2 User requirements processing 82
5.2.1 Determine the DOF of wrist joint 82
5.2.2 The processing of user fuzzy requirements 84
5.3 Service-component configuration 86
5.3.1 Similarity and relevancy calculation of service-component 86

5.3.2 Service-component selection and instantiation 89
5.3.3 Service-component object selection 90
5.4 Configuration program evaluation 91
5.5 Case study for user-customized configuration design 94
5.6 Chapter summary 98
CHAPTER 6 DESIGN PLATFORM PROTOTYPE SYSTEM DESIGN AND APPLIED RESEARCH 100
6.1 System Requirements Analysis 100
6.1.1 Use case diagram for platform 100
6.1.2 Role-based business process analysis 101
6.2 System design 103
6.2.1 Function module design 103
6.2.2 Platform physical configuration design 105
6.2.3 The platform architecture 106
6.3 Platform page design 107
6.3.1 Platform home page 107
6.3.2 Product releasing and display 108
6.3.3 User-customized configuration design page 111
6.4 Chapter summary 116
CHAPTER 7 SUMMARY AND OUTLOOK 117
7.1 Research summary 117
7.2 The main innovation 119
7.3 Work prospects in future 120
REFERENCES 122

CHAPTER 1 INTRODUCTION

1.1 Background

Industrial robot is a type of mechatronic product containing multiple fields' components such as machinery, electronics, control unit, and computer. It is characterisedby complex structure, long lead times, and high manufacturing costs. Currently the industrial robot technology focuses on modularization, reconfigurable robot, intelligentization. The main research of these technologies is to design the industrial robot to modular structure, and then reconfigure these modules to new industrial robot with new features. The modularization technology allows the production of industrial robot to achieve mass customization, and it solve problem of the long development cycle and high development cost to a certain extent. However, an industrial robot is required to achieve high degree of customisation for finishing special work. In the increasingly fierce global competition, only more rapid respond speed for diversified customer requirements can make enterprises to be invincible.

To achieve high degree of customisation, customer participation configuration design is more and more going into people's attention. The mature application is the component technology in software field. In the field of hardware products, the most widely used product is the computer. All parts and components from different manufacturers in the computer hardware field have achieved standardized interfaces. Customers can choose graphics cards, hard drives, monitors, etc. at different prices and performance according to their needs, and then assemble these parts and components into a fully customized computer. Since all the interfaces have been standardized, customers can ensure the feasibility of the final product while not having specialized computer knowledge. The technology thanks to the rapidly unified industry standard of computer industry. Although certain manufacturers' interests are suffered in short term, it promotes the rapid development of the computer industry in long term, and each enterprise of the computer industry can benefit from it.

As a complex mechatronic product, it is more complex for industrial robot to take customer participation configuration design than a computer. The vibration, deformation and other issues of configuration product during operation process are likely to lead to the failure of design. Even the car which now has begun customization, such as Rolls-Royce, Volkswagen, etc., it only provides customers with simple customization service such as pedal customization, seat piping customization, etc., and it can't provide a deeper level of customer participation

configuration design. But the basic for industrial robot to achieve customer participation configuration design already exists. On the one hand, the current industrial robots are mostly modular structure, and there are a lot of researches on reconfigurable robots, which are the prerequisite for customer participation configuration design. On the other hand, industrial robots are not the same as other mechatronic product such as automobile. Automobile also has modular structure, but its function is complex, many modules are required in one product, and the majority end-users do not have the professional design knowledge, so it is difficult to implement user-customized. But industrial robots are usually used in factories to do some simple and repetitive work, so a single function is required. In addition, usually the user of industrial robots is factories, which have employees with mechanical expertise. These characteristics provide the possibility to realize a deeper level of customer participation configuration design. Therefore, the industrial robot can also realize fully customization through customer participation configuration design like the computer, which can promote the rapid development of the industry.

Based on the above analysis and the existing research, this thesis proposes user-customized configuration design pattern for industrial robot. It is not just a simply sentence that changing the industrial robot development approach from professional designer design to customer configure design. The changes of development approach bring the role changes of relevant enterprises such as industrial robots manufacturers, parts providers and developers in industry chain. To ensure that customers can design industrial robot by their own, a clearly specified set of business mode is needed to manage all stakeholders. On the other hand, design activities performed by the user don’t use the physical components in the real world, but use the virtual components in computer. Therefore, how to transform the physical components into virtual components, how to provide non-professional designer - user with a simple design method and design process, and how to validate the user's design result to ensure the feasibility of the configuration result become the key issues in user-customized configuration design pattern. Aiming at these key issues, the thesis does research in depth.

1.2 The research status and analysis

1.2.1 The research status

There are three relevant areas to be studied: the first one is that in user-customized configuration design pattern, the product is ordered by the user

online and receipted and used offline, so it is needed to know relevant business mode of online and offline integration; the second one is that in user-customized configuration design pattern, the design process is to configure the predefined components to product in mass customization production mode essentially, so it is needed to know the relevant design theory in mass customization production mode such as modular design, product family design, customer participation configuration design, reconfigurable modular robot design, etc.; the third one is that in user-customized configuration design pattern, the components want to realize the function of fast plug and play like software components, so it is needed to know the plug and play technology in component technology in software engineering. The research status of these related issues are introduced as followed.

1.2.1.1 The business mode of online and offline integration

Paul Timmers believes that the business mode is an operating mechanism about enterprise product flow (service flow), capital flow, information flow and value creation process, and it consists of three elements: (1) the participants' status and their roles; (2) the enterprise benefits and income sources obtained in the business operation; (3) the value created and reflected in the business mode [1]. With the rapid development of e-commerce in recent years, the traditional business mode with only physical stores has been greatly impacted. More and more companies try to carry out the business mode of online and offline integration. The common modes include B2C (Business to Customer), C2C (Consumer to Consumer), B2B (Business to Business), O2O (Online to Offline).

B2C means that the seller online is enterprise and the buyer is individual, namely the consumer. B2C is actually moving the store to online. It shows and sells goods to consumers via the online store, and the enterprises do not directly face to face with customers. Its operating mode is shown in Figure 1-1 [2]. Consumers search products by B2C online store, and then create order through online payment after searching the right products, so the product supplier delivers the product to the consumer through the logistics company offline. As the buyer of B2C mode is individual, so this mode is generally used to sale product of common household items or services which are closely related to the personal life. The bulk product of industrial field or highly specialized products generally don't use this mode. For the website in B2C mode, how to obtain a higher inventory turnover to reduce costs and how to improve on-time delivery rate to improve customer satisfaction rate is the key to succeed in the fierce

competition [3].It also needs to improve the website management quality to attract customers [4]。C2C means that the seller and the buyer online is individual, and the website only offers a trading platform. Its operating mode is similar to B2C, and the only difference is the identities of the participants. C2C and B2C both sell products to individual consumers. Since any individual under the C2C mode can be the seller, its openness leads to a huge advantage on the merchandise source and type. But the advantage of this mode also brings a disadvantage-because the personal behavior is unsupervised, the seller may sell inferior goods and take false trading which makes buyer believing that the quality of products. There are research on how to identify seller fraud to restraint the behavior of buyer and seller now[5].The buyers and users of industrial robot are manufacturing companies generally, so far in the field of industrial robot seldom exist B2C and C2C mode.

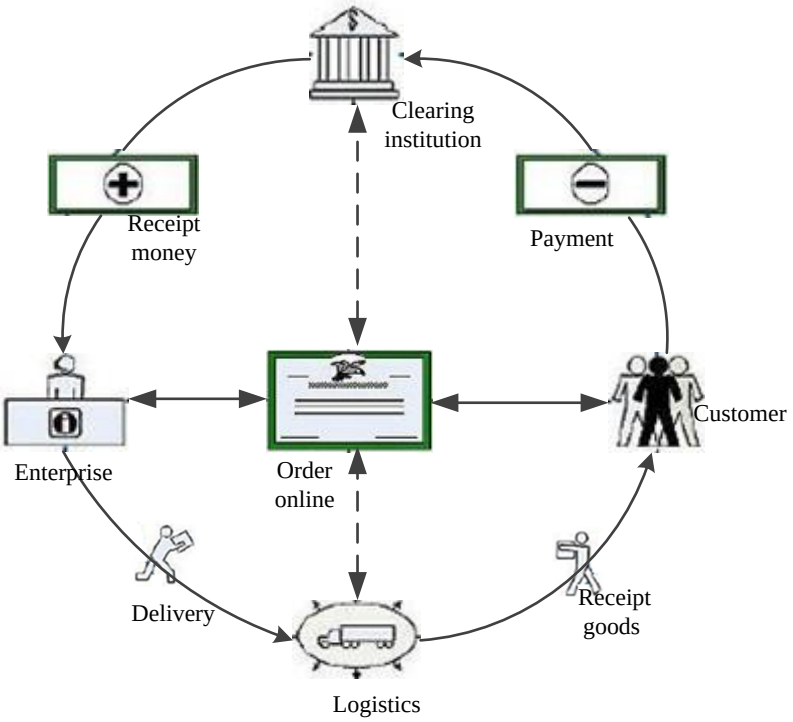


Figure 1-1 B2C mode [2]

B2B means that the seller and the buyer online is company. Its operating mode is similar to B2C. In a variety of e-commerce business mode, B2B occurs first. From the beginning of the 1990s a number of B2B websites were established and became the leader in the industry. For example, in industry field as the more famous B2B websites include the *GlobalSpec* established in the United States in 1996, the *Thomas Global* in the United States in 1995, the world's first enterprise directory *Switzerland kompass* published in 1945, the Directindustry established in the France in 1999, etc..

For the existed problems in application, many scholars do related research, such as determining customer-oriented performance improvement approach through studying the sales staff behavior of the website [6], researching how trust affects B2B by studying the relation between the social trust and B2B [7], researching how to apply the B2B theory in industrial field for the features of industrial product [8], researching Brand Management in industrial field B2B [9], researching how to use B2B as a new distribution channel through the distribution channels analysis of traditional entity companies [10], etc. Currently B2B business mode has matured. Users can not only inquire and purchase product online, but also can inquire the industry standard, inquire tradeshows information, and display industry brand directory and product directory. It can establish a website integrating the functions of inquiry, display, trading for global buyers and suppliers.

The concept of O2O is formally proposed on *TechCrunch* by Alex RamPell of American in 2011 [11]. The core content is to pay online and get goods or services on physical store offline. Different to the previous three modes, O2O mode has physical stores, and customers need to pick up products or services in the physical store after the payment online. For customers, payment online can enjoy more favorable price than payment offline. The literature [12] researches how to achieve the combining of pricing offline and promotional activities online. For physical stores, the products can be promoted i via online websites, and the dependence on physical store location can be reduced. Simultaneously, each transaction has a detailed record to facilitate the analysis of market trends. Meanwhile according to Daniele Scarpi et al study, customers not only purchase products online, but also want to get more fun online[13].Therefore online websites can consider to provide customers with more fun in premise of meeting the basic transaction, so as to promote sales of the products offline. Most industrial robots companies currently only have physical store offline, but the famous industrial robot company ABB has established its first O2O experience store in September 2014. It achieves the O2O business mode of "experience offline, transaction online, payment online, delivery offline", and becomes industry leader of O2O mode in robot field.

Each business mode has different applicable object. For example, the B2B is applicable to bulk industrial products, the B2C and C2C is applicable to groceries, the O2O is more suitable for restaurants, theaters and other products which need to enjoy services in physical store. The user-customized configuration design involves

purchasing parts online (B2B or B2C), transaction design results online (B2C or C2C), enjoying assembled products or service offline (O2O). It is needed to combine various kinds of business modes, and study business mode applicable to user-customized configuration design.

1.2.1.3 Related design theory of mass customization production

Mass customization (MC) is considered to be the most important production mode in the 21st century [14]. By computer technology, modern design methods, flexible manufacturing technology and so on, it can produce customized products meeting different needs of customers on the premise of closing to the cost advantages of mass production [15]. Mass customization technology is mainly achieved through modular design, product family design, configuration design.

Modular design

Modular design is a modern design method. The main principle is to break product down into sub-components, i.e. product modules. This division can promote the standardized of components and increase the diversity of products [16].In the 1950s, some European and American countries have officially put forward the concept of modular design. After decades of development, the modular design has very mature theory, and the research contents mainly include module division and recognition, module management [17], module evaluation and optimization. The module division method is the most critical issue. It is needed to consider what extent of modularity is optimal and what product information is needed in module, etc. [18].To solve these problems, there are methods such as the explanation structure model-based method [19],Design Structure Matrix-based method[20][21][22],fuzzy clustering-based method [23], method based on the FBS(function-structure-behavior) mapping [24], self-organizing maps method[25], method based on triangle algorithm figure [26], the biggest smallest division method[27].The module evaluation and optimization methods contain genetic algorithm-based method [20][28][29][30], module optimization based on simulated annealing algorithm [31], fuzzy comprehensive evaluation method based on multi-experts [32].Based on mature theory, modular design has been widely used in the computer, automotive, machine tools and other products. Most of the products in these areas have achieved a modular structure, and they can change product features flexiblely by adding, deleting or replacing certain modules.

Product family design

In product design process, product family design no longer design a particular product alone, but t design a set of products with similar functions to meet different customer needs. The effective way to achieve product family is to classify customer needs and define product platform according to the common needs. Platform-based product family design is an effective way to achieve mass customization [33]. The product platform is a generic and shared modules collection, and the product family is a set of prducts with similar structure derived from the product platform [34]. Here the concept of module is an physical concept, meaning a replaceable unit with separate function, and the interdependence between modules is the smallest [35]. The condition of one component becoming a module is: the functions, space, and other interface features of the component exist within a particular interface rang of the modular product [36].By platform-based product family design, the reuse of product design results can be achieved, so that enterprises can respond quickly to customized needs of different customers in the context of low-cost, short development cycle. The design method to achieve platform-based product family design can be divided into two types: (1) module-based product family design. The product diversification is achieved by adding, replacing, removing the product platform modules [37][38]. (2) parameters-based product family. The product diversification is achieved by magnifying, shrinking and other operations the product platform modules [39][40].In the actual implementation process, two methods are generally used in combination in order to improve the flexibility of the product platform.

For the platform-based product family realization, the main researches focus on product family modeling and product family design optimization. For product family modeling, there are multiple view product family model [41][42], product family structure model based on common modules [43], product family model based on alterable module [44],product family model based on GBOM [45][46],product family design model based on biological DNA [47]. For product family design optimization, the method can be divided into two categories: single-stage optimization and multi-stage optimization [48]. The method based on module attribute assignment and selection introduced by is a single-stage optimization method [49]. The optimization method based on multi-objective genetic algorithm [50][51][52], the optimization method based on population genetic evolution mechanism [53],the optimization method based on rough set and fuzzy set [54], the selection method of modules based on simulated annealing [55], are all belong to multi-stage optimization method. The

multi-stage optimization method avoids the difficulty of solving when the product family model is too complicated. Regardless of the method using single-stage or multi-stage method, all the alternative solutions designed by the designers are needed to be evaluated after the formation. Tarek propose a universal method based on physical components, which can automatically generate re-designed solutions [56]

Configuration design

The configuration design is to find those configuration results that meet all the requirements according to the predefined components collection, configuration requirements description and configuration standard, and then product customization is achieved [57]。Product configuration design is an effective method to solve contradiction between the diverse user demand and the product development time and cost [58]. In recent years, most of the researches related to product configuration focus on product configurator. Product configurator is essentially a knowledge-based expert system that allows users to select certain parts from the predefined set to consist of effective products [59]。Lars Hvam in Technical University of Denmark studied a development method for visual configuration system[60]. First he introduced what visual configuration system was, and then divided visual configuration system development into seven steps: process analysis, product analysis, object-oriented analysis, object-oriented design, programming, implementation, maintenance. Manufacturing enterprises could configure product formally in accordance with the method. Sometimes product platform based on module can’t meet some special features of product. In order to solve this problem, Sehyun Myung and others studied parametric design in the configuration design [61]. They put forward knowledge-based parametric design method of mechanical product. This paper presented the concept of design unit, and designed an expert system framework of parametric design based on design unit and configuration knowledge, which integrated the configuration design knowledge and detailed knowledge of CAD systems. T.W. Carnduff proposed a model of configuration management, which managed configuration process through version evolution [62]. Configuration version information was captured flexibly at the component level and stored in the database. Designers did not need any knowledge of programming techniques. They could use the model to track the configuration evolution, then selected appropriate configuration version and module through graphical user interface. In article [63], authors developed a desktop virtual reality-based interactive modular fixture configuration

design system. The authors used multi-view furniture assembly model to aid in information representation and management. The model could express information view of fixture module, functional / structural information view, assembly relationship view, and 3D operation methods was used to accurately locate the position of fixtures in the three-dimensional space. These studies provide a solid theoretical foundation for the implementation of configuration system.

The key issues in the implementation process of the configuration system to be addressed include configuration platform expression [64][65], configuration design implementation method, configuration solution evaluation and the optimization [66][67]. Wherein the configuration design implementation methods contain case-based reasoning method [68][69], product structure model-based method[70][71], knowledge-based method [72][73], etc.. The research emphasis is configuration model and configuration algorithms, and there are rare researches on configuration design management from a global perspective. With the growing research on configuration design, F.Salvador et al. found that only solve the problem of designing a configurable product family and combining predefined components into effective products is not sufficient to reduce significantly the trade-off between customization and delivery time. They investigate 122 product manufacturers using configuration design method, and the result is shown in Figure 1-2 [74].

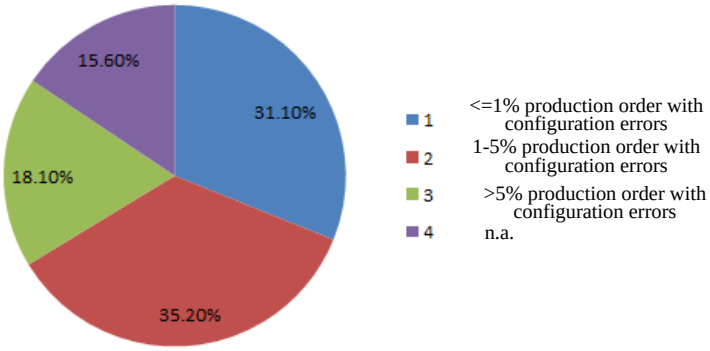


Figure 1-2 Percentage of configuration errors in companies [74]

Even with the configuration design method, some problems still occurs because of the lack of communication between companies and customers, which will lead to run counter to the initial targets of customization and rapid delivery. In order to resolve this contradiction, recent years' research gradually turn to design product configuration process from management view and holistic view, and extend the management from product configuration design to a uniform framework including users, designers, suppliers, salesman [75][76].

Customer participation design

Customer participation design is actually a design method for mass customization production. Because it is very difficult for the customers to design a completely new product, and therefore more customers configure the existing components to product. Randall et.al. pointed out that the product customization needs to meet the five principles, including providing different customized interfaces for different users, providing customized starting point, providing modify space, providing design prototypes for customer design reference [77]。The most critical issue is to give customers a product structure template as a basis for the customer participation design. For example，Roger and others proposed a generic product family master model [78]. Based on web service, the model integrated product life cycle information including design, manufacturing, supply chain and so on. The model could provide support to custom-configure of products. In literature [79], product structure model was defined as a multi-layered product functionality / behavior structure class system. It was expressed by the composite tree, and not contained a specific product or component. Product configuration process was a process of instantiating product structure model. The author also introduced the concept of virtual modules. The expansion modules which is not existing but the MC enterprise could design and manufacture were introduced into the configuration model. In literature [80], the definition of the product template was set first, then the ordered tree model of product configuration was built. On that basis, the product configuration process was analyzed. Rapid product configuration design based on product design template was studied in literature [81][82]. Different methods were used to build a product configuration template in the two literatures. For specific industrial robots product, literature [83] provided a heterogeneous modular structure, including a slider module, scalable module, rotation module, hinge module as the basis or toolbox for the industrial robot development. Literature [84] developed a heterogeneous toy robot platform. Customers with non-professional knowledge can configure different toy robots using different modules based on their own interest

In practical applications, the customer participation design first appeared in the service, clothing, furniture industries, and later the most widely used in the computer field. For example, the Dell computer of United States provides customers with complete customization capabilities. Customers can choose the right components on its website according to their needs, and then the components will be assembled to a

personal computer by Dell company. Due to the composition structure complexity and the high manufacturing costs of the mechanical products, less customers participate in the design. But in recent years with the development of industry technology, there have been customization for large-scale mechanical products, such as the US trucking company Paccar, which achieving customized mode. There will not be two identical trucks in one production line.

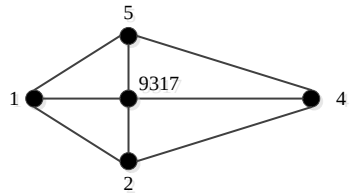
1.2.1.3 Reconfigurable robot technology

In1988, the first reconfigurable robot was developed at Carnegie Mellon University [85]. Followed by many Europe and America countries have begun to study reconfigurable robots. Reconfigurable robot is composed of a set of exchangeable modules which have different sizes and performances, and it can be assembled into a variety of different configurations in order to adapt to different required work [86].

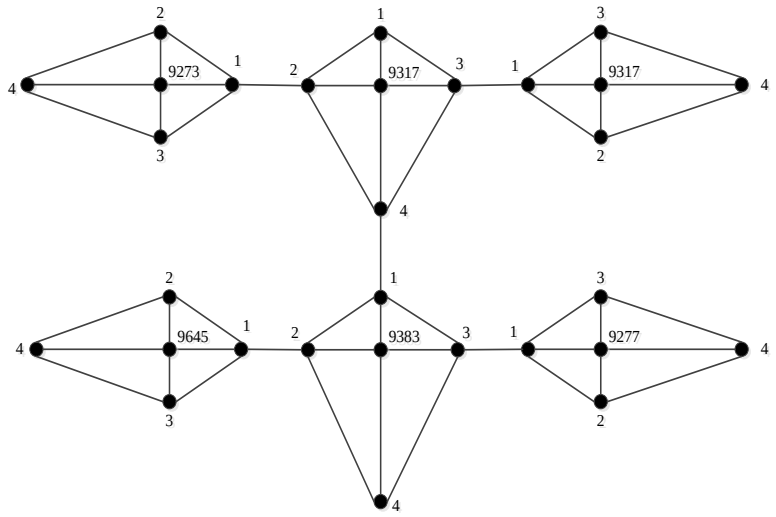
Reconfigurable robot is generally realized by a modular design approach. Modules have a certain autonomy and perceptual skills, and each module has a uniform interface for transmitting force, motion, energy, and communication [87]. Reconfigurable robot can change configuration by the help of outside force. In recent years, more research is self-reconfigurable robot, which is the extension of reconfigurable robot. It can change its configuration independently without outside help. In the self-reconfiguration process, the most critical issue is how to express robot module to complete configuration changes. The literature [88] expresses the robot module as shown in figure 1-3 (a). 9317 represents the module unique number. 1234 represents the four faces around the module. The three-dimensional module can be transferred into a plane graphics by figure 1-3, and then the correlation matrix is used for the plane graphics transformation. One of the configuration using the plane graphics modules is shown in figure 1-3 (b). The literature [89]divide the industrial robot modules into rotation (Rotary, Swivel), Translational, Link, as represented in Fig. 1-4 (a). The different configurations composed of these modules are represented as Fig 1 - 4 (b). These methods have a strong focus on the module structure information, but not expressing specific motion parameters information of the module.

In addition to the module expression, the research contents of reconfigurable robot also include module division and platform building oriented by reconfigurable robot[90][91], module configuration [92][93], the reconstruction design methods of

modular robot [94][95][96], dynamic and kinematic modeling [97][98][99], self-reconfigurable robot motion planning [100]. These literatures research reconfigurable robot from the different stages of the configuration design theory, including the previous module division, module expression after division, the reconstruction method from module to robot, as well as the later dynamic performance solving, etc. These researches provide a solid theoretical foundation for robot configuration design.

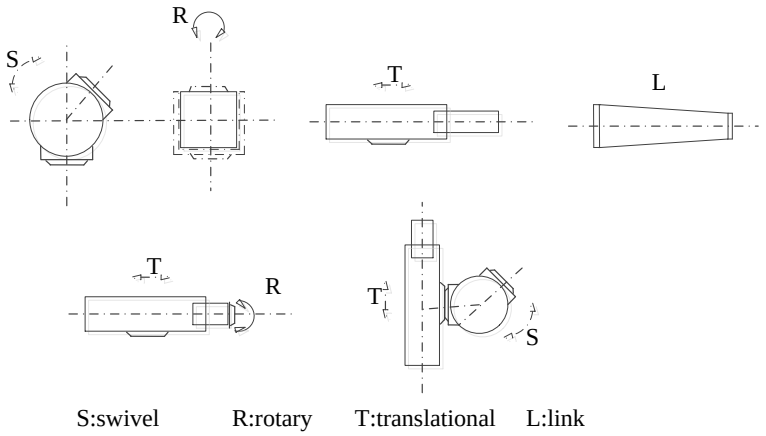


(a) Module expression for robot

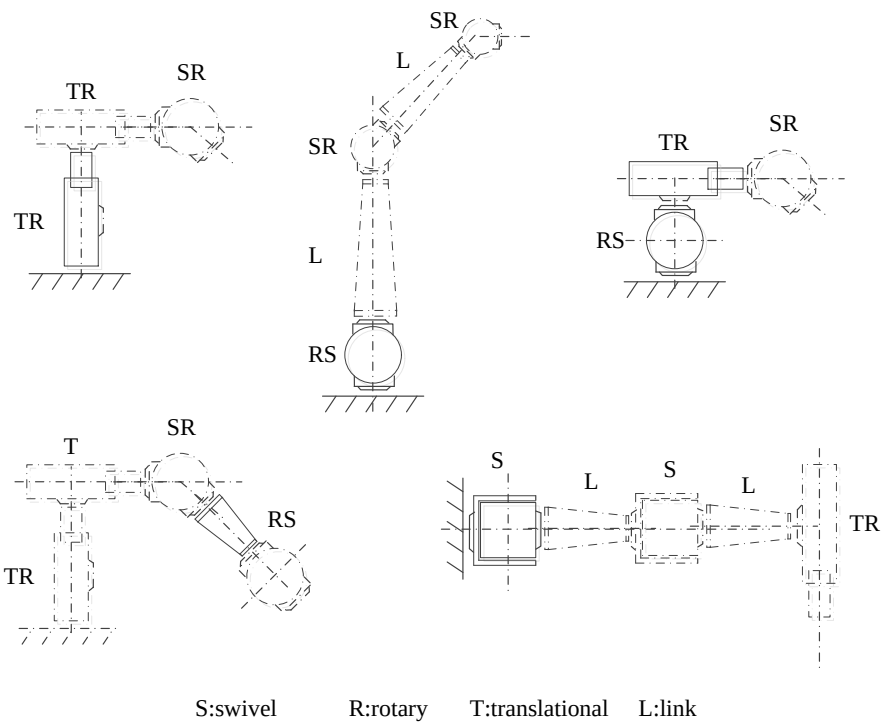


(b) Robot configuration

Figure 1-3 Homogeneous robot expression method[88]



(a) Standard industrial robot modules



(b) Different configurations of industrial robots

Figure 1-4 Expression of industrial robots [89]

Currently, many research institutions have been developed reconfigurable robots, such as the HexBot of Sharjah University [96], SuperBot of Southern California University [101], YaMoR (Yet Another Modular Robot) of EPFL(Ecole Polytechnique Fédérale de Lausanne) [102]. These reconfigurable robots are all homogeneous system which is the same as shown in figure 1-3, and the modules are basically the same type. Also there are many heterogeneous robots like figure 1-4, such as Recrob of Technical University of Cluj-Napoca [103], SMART of Universidad Politécnica de Madrid [94], Odin of University of Southern Denmark [104], etc.

Industrial robot belongs to robot. Since 1962 the United States developed the first industrial robot, many countries have started the design and development of industrial robot [105]。The research on reconfigurable industrial robot are more concentrated in the practical applications, and there are many industrial robot enterprise with mature technology worldwide, such as FANUC and YASKAWA of Japan, American Robot and Emerson Industrial Automation of America, ABB of Sweden, KUKA of Germany, COMAU of Italy, etc..

The industrial robots of manufacturing companies have been realized in a

reconfigurable modular structure. But due to the non-unification of the upper standard and due to technical security in companies, industrial robots can only achieve modules reuse within the same company, and cannot achieve matching across different companies. This factor limits the user's configuration design.

1.2.1.4 Component technology in software engineering

Component technology is evolved from the object-oriented design. Object-oriented design is an systematic approach using the object-oriented idea to software development process to guide software development activities, and the constituent elements include classes, objects, encapsulation, inheritance, polymorphism, messaging [106]. In this approach, the objects in reality are mapped to the software objects. The real world is reflected by building links and interactions between the objects. Component technology is a further development of object-oriented technology. First component is a program body with specific function, and it can be able to work independently or be assembled together with other components [107]. Component technology packages the program as component, and provides the user with services through the interface hiding implementation process. Through reuse existing components, component technology reduces the complexity of software products, and shortens the development time [108]. Currently, the components framework modes include COM (Component Object Mode)developed by Microsoft, COM+, Distributed Component Object Mode (DCOM), CORBA (common object Request Broker Architecture) for UNIX developed by the Object Management Group and Java Beans [109]. The study includes components development and description for different areas [110][110][111], component obtaining[112][113], component management and maintenance[112][114], component evaluation[115][116][117].

As a leading-edge technology for software engineering development, component technology has been widely applied to different areas of software systems. The more applications in industrial is the configuration software and open CNC system. The concept of configuration is a terminology from the computer field [118]. The implication is that the use of component technology to develop software that allows users to configure the needed software system using components. Then the configuration starts to use in the industrial areas, and the software who can configure the industrial process monitoring and control system quickly for industrial users is called configuration software. Currently, there are many configuration software

companies, such as the INTouch and Fix of United States, Citech of Australia, WinCC of Germany [119].With the increasing requirement for automation, customization and integration, the configuration software needs to be integrated with more discipline, such as the fieldbus [120], ethernet [121], distributed systems [122]. Open CNC system is a standardized, replaceable, scalable, reconfigurable hardware and software systems, and the system modules have plug and play compatibility [123]. Academics have been carried out in-depth study for the realization of open CNC system, such as the cross-coupled controller of University of California [124], distributed control systems based on the IEC 61499 standard of the University of Auckland in New Zealand [125]. In practical applications, the Ford, GM, Chrylser and other companies of United States have jointly developed an open and modular architecture controller for the automotive industry; three machine tool companies (Toshiba Machine, Toyoda Machine Works, Yamazaki) of Japan and three information systems development company of Japan (IBM, Mitsubishi Electric, SML) jointly proposed a hierarchical model for controller. Modules of each level have a clear service contents and interface specifications; the OSACA (Open System Architecture for Control within Automation System) program proposed by European countries developed a manufacturer-independent architecture control system [126]. As can be seen, the study of the component technology is already very mature in theory, and also has been widely used.

1.2.2 The research status summary and analysis

Currently, for the configuration design, reconfigurable design of hardware products, and component design of software product are all focus on the product modularization and modules configuration to reuse the design results. On this basis, how to further push forward again to achieve customer directly participating in the configuration design is currently one of the hotspots. Due to industrial robots involves many fields, the configuration of the product can’t be put into use immediately. It needs to ensure that the kinematic and dynamic performance, resonance, accuracy of the configuration meet the criterions, and therefore the appropriate way needs to be taken to ensure the success of the design results.

Scholars explore the solutions of industrial robot configuration design from different technical point of view, and also give the necessary attention on customers directly participating in the design process. These researches enhance the rapid response capability of industrial robots, but there are still some problems:

a) Modular industrial robot is the basis for configuration design, but the existing research on the industrial robot modules have stayed mostly on specific product, and the description of the module focuses on the specific structure. Configuration design not only includes configuring the product structure, but also can be able to characterize the module structure, kinematics and dynamics parameters in the system layer;

b) The researchers studied on robot configuration independently and were limited to technical details. But the crucial point of the configuration design success is to have enough modules to choose. How to develop a unified module standard, integrate the standard modules together to configure, and study the entire configuration design process from a management perspective for industrial articulated robots?

1.3 Thesis significance and main research contents

1.3.1 Research significance

Combining the product configuration design theory with characteristics of the industrial robot, the thesis researches the user-customized configuration design method for industrial robot. Firstly, the business mode of user-customized configuration design is given, and the mode is simulated. Then the key technologies to realize product design platform are in-depth studied, including component expression, configuration template design, configuration internal algorithm design and other issues. The design platform integrates various types of participants (including customers, suppliers, manufacturers, etc.), allows customers to match, retrieve, and optimize the components according to personalized requirements. Platform will allow more companies to participate in the design and manufacturing engineering, while also reducing design iterations. It solves the contradiction between diversification requirement and manufacturing cycle and cost. The study provides a new approach for modern product design. The research is not only used in design and management for industrial robot, but also can be extended to various types of mechatronics products, such as machining center, automobile, aircraft and other products, which having a wide range of practical applications.

1.3.2 The structure of research contents

Around the issue of how to implement user-customized configuration design, the thesis do research mainly from four aspects, including user-customized configuration design pattern introduction and simulation, industrial robot component expression, user-customized configuration design template building, internal algorithm analysis. Based on these four issues, the thesis is divided into seven chapters, and the relationship and the role of the main chapters are shown in Figure 1-5. The core chapters are from chapter 2 to chapter 6, which describe the key issues needed to be addressed.

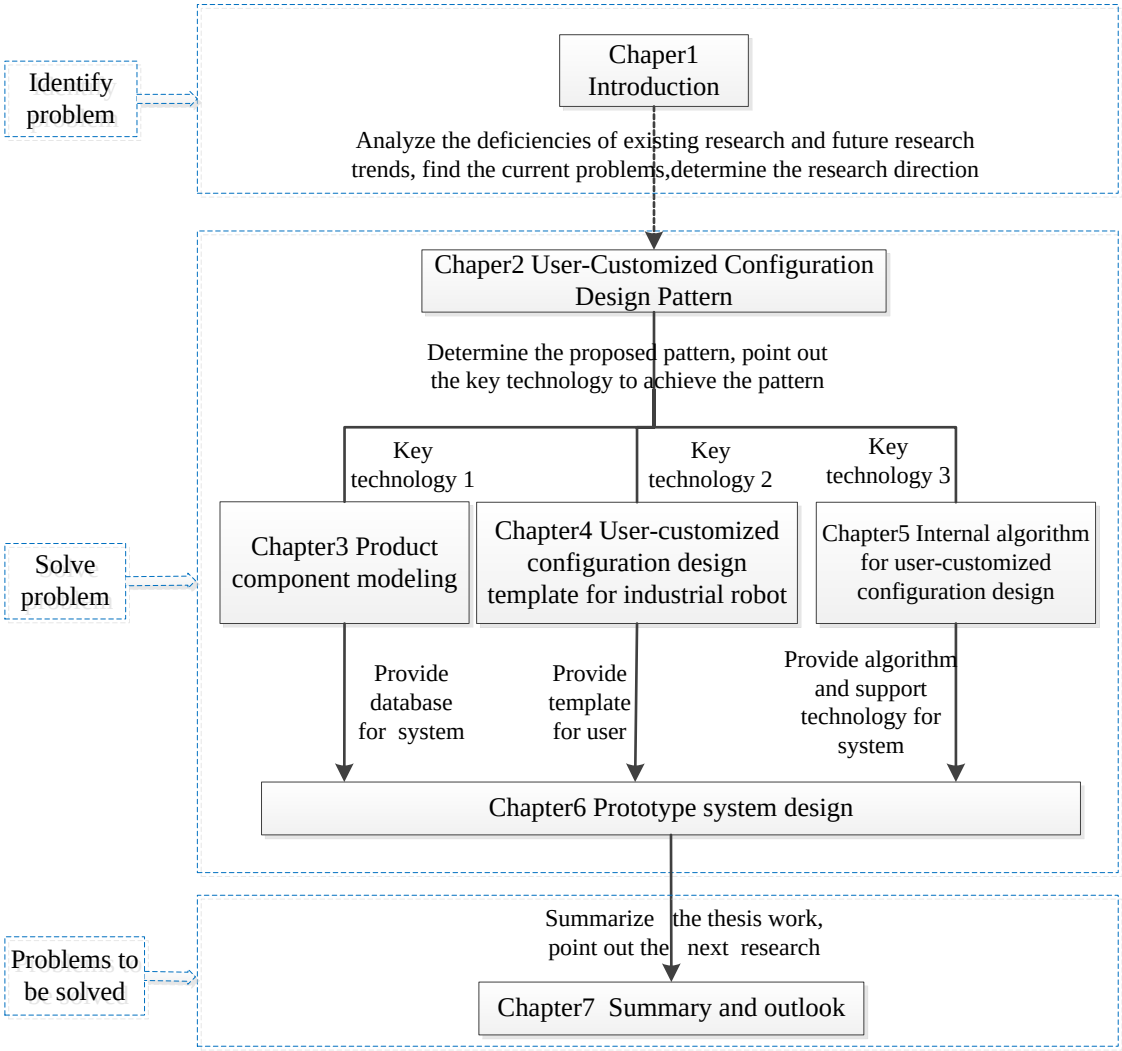


Figure 1-5 The overall structure of the paper

In chapter 1, research direction is determined by studying the status analysis. The research status of design method such as modular design, product family design, configuration design, reconfigurable robot design, object-oriented design of software, component-based design of software, and the ontology-based knowledge

representation is summarized. By analyzing the deficiencies of existing research and future research trends, the research direction is determined combining with the current problems faced by industrial robot design.

In chapter 2, the user-customized configuration design pattern is put forward in view of the characteristics of industrial robot. The related concepts are expounded, including pattern, configured product, component, service-component, user, user-customized configuration design, design platform, etc. Then the corresponding business mode of user-customized configuration design pattern is given according to business mode elements, and the key technologies to realize business mode is studied. Finally, system dynamics models were established for user-customized configuration design business mode and for traditional business mode of industrial robots by Anylogic simulation software.

In chapter 3, the multi-field components in the real world composing an industrial robot is formalized. Object-oriented thinking and ontology is analyzed. On this basis, the service-component is introduced. Service-component is used to describe component. Service-component is composed by the ontology layer, object layer, management layer. And a description method case of service-component is illustrative, which shows that the service-component can comprehensive describe the multi- field component of industrial robots.

In chapter 4, the user-customized configuration design template for industrial robot is built. The industrial robot user-customized configuration design template is constructed through analyzing the industrial robot composition structure. The kinematics and dynamics analysis on template is taken by Simscape model. The finite element analysis on template is taken by ANSYS, including statics analysis and modal analysis. And the configuration template is analyzed in application.

In chapter 5, the internal algorithm for user-customized configuration design is provided. The platform-based and user-leading user-customized configuration design process is studied. The process gives the operations of user, the tasks finished by platform, the model and database needed by platform. Then the internal algorithm to keep the design running smoothly is studied, including the degree of freedom determination, fuzzy demand calculation, and service-component configuration. Finally a case analysis is taken for the internal algorithm.

In chapter 6, a prototype system of product design platform on the basic of the theory study is designed. Firstly, system requirements analysis, system design is

carried out. And then the main interfaces are designed.

In chapter 7, the main research work and innovation of the thesis is summarized, and the next work is prospected.

CHAPTER 2 USER-CUSTOMIZED CONFIGURATION DESIGN PATTERN

In general configuration design pattern, the user is responsible for proposing requirements, the designer is responsible for product family designing based on requirements, and then configure the product family into customized products based on user' requirements. In user-customized configuration design pattern, the final configuration is not completed by designer but completed by the user according to their own needs. In this pattern, the designers release basic functional modules via a configuration platform, the users configure product on the platform, and the platform ensure the product effectiveness configured by users. Through this pattern, the iterations between designer and user can be reduced, and more companies can participate in the design and manufacturing process, which facilitates the development of industrial robot industry.

2.1 Theoretical Basis

The user-customized configuration design pattern drew on service-oriented architecture (SOA) of software development. As stated in the introduction, software development has experienced the process of process-oriented development, object-oriented development, component-based development, and service-oriented architecture. Each kind of method has a higher level of abstraction and reuse than the former method. The industrial robot design method and software development method learn from each other in the development process, and there are a lot of similarities as shown in figure 2-1.

Process-oriented software development is programming by functions, similarly, the initial product design method design product according to the required functions and the implement process of the function.

In order to improve the efficiency of programming, object-oriented development method is put forward. The real thing is abstracted into object, and programming efficiency could be improved by object reusing. In the mechanical products design, in order to reuse the previous design result, designers also commonly apply standardization parts structure directly to the design process rather than starting design from the basic functional analysis, which actually treat parts as the object, and complete design through to the object reuse.

Either in function flow design or standardization parts design, the result gained from design activities is a complete set of design drawings. The product seen by user is a complete product manufactured according to the design drawings, but they can't see the internal structure of the product, also don't know how the product is design. Similarly, in process-oriented and object-oriented development, the analysis and design result is a complete programming. What the user sees is the software product through running the programming, but they can't see the internal programming structure of the software.

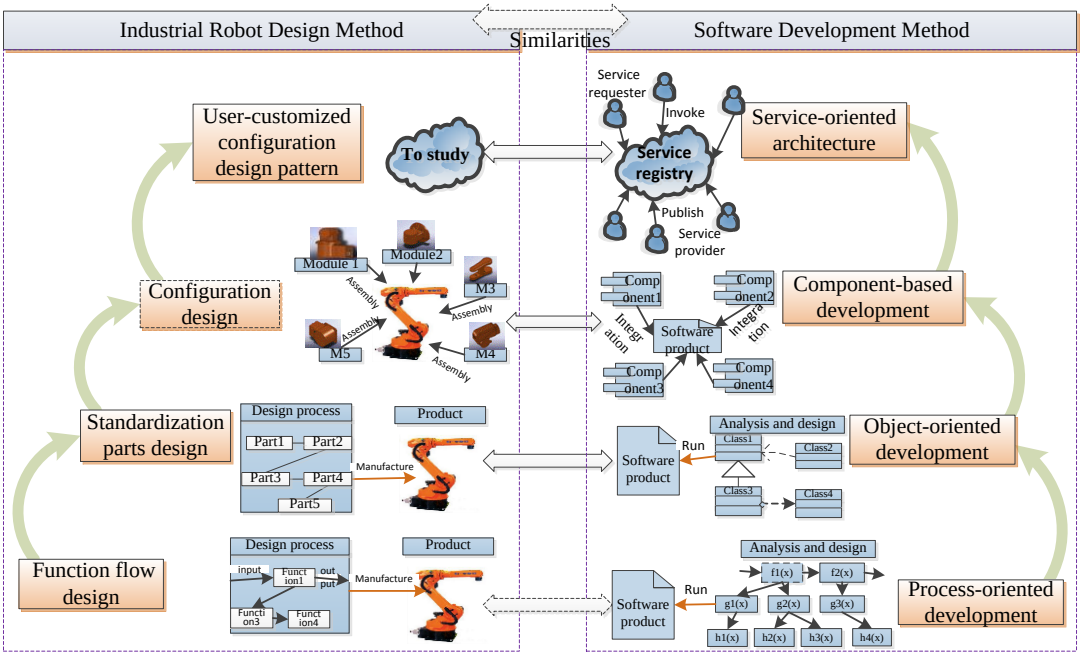


Figure 2-1 Similarity comparison between industrial robot design method and software development method

In order to achieve a higher level of reuse, component-based development method is put forward. Object belongs to the concept space, but the component belongs to the physical space, and it is a physical part of software product. The component encapsulates the implementation process, and only interacts with the outside world through interfaces. Through assembling different components, different systems could be formed. Component-based development liberates software developers from the programming. It allows software developers no longer study at the low-level details of programming, but focus on how to integrate the components to the required software system. Unlike the process-oriented and object-oriented development, in component-based development method, what the user sees is not a whole software product, but a software product assembled by different components. Users can clearly understand the structure of software product, and also can delete or

replace a component without affecting the entire product. Configuration design method of industrial robots is similar, which encapsulates the parts and components with specific function to physical modules with standard interfaces. Designers no longer focus on design drawing of specific parts, but more focus on reusing previous module and assembling modules into the product. What the user sees is no longer an integral whole product, and the product can add, remove, or replace one module according to need.

Service-oriented architecture is based on component-based development. It does not study at how to develop a specific software product, but makes demands on software development from macro architecture level. Under this architecture, different components can be encapsulated as service, and then be published to the network. The software developers complete their business through accessing services distributing in different server. In this architecture, everyone can publish services as a service provider, and also can invoke the service of others as a service requester. At present the modular design and configuration design of industrial robot already has a mature research. Therefore, compared to the service-oriented architecture in software development, the design method of the industrial robot can also be a step forward. However, as a kind of mechatronics products, the industrial robots module reusing can’t be achieved by simple network access, and it needs manufacturing, logistics etc. in the real world. Also the modules configuration can’t be achieved just by interfaces integration, but need to consider the mechanical performance, electrical performance, kinematic and dynamic performance etc. in the configuration process. Due to these differences between software product and industrial robot, it is infeasible to simply imitate the service-oriented architecture of software development. Based on this, the thesis propose user-customized configuration design pattern of industrial robots to guide the industrial robot design.

2.2 Related Concepts of User-Customized Configuration Design Pattern

Before giving the definition of user-customized configuration design pattern, the concept of pattern should be clarified. A classic definition was given by Alexander et al.:Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same

way twice[127]. A design pattern is a solution to a recurring problem [128]. In general, design pattern is used in software engineering to solve problems in code design. There are two key elements in the concept of design pattern: problem and solution. Based on this, user-customized configuration design pattern is defined as follows:

Definition 1 (User-Customized Configuration Design Pattern): In the process of industrial robot configuration design, **[Problem]**the rework of design often arises because of not meeting customer's requirement completely. To solve this problem,**[Solution]** a product platform is provided, designers publish basic components to the platform as service-component (SC), component manufacturers select appropriate component to manufacture, and users customize product by invoking SC on the platform according to their own needs. This solution to solve the problem is called user-customized configuration design pattern.

In this pattern, the final configuration is not completed by the designer but by the user according to his/her own needs. Through this pattern, the iterations between designer and customer can be reduced, and the configuration design time can be shortened. Through the product platform, all stakeholders of industrial robots can be integrated to achieve and validate the whole configuration design process efficiently.

When not specified, the product mentioned in the thesis refers to industrial articulated robot. The related concepts in definition 1 are introduced below. The first one is the concept of component. In mechanical product, the component is composed of parts. In user-customized configuration design pattern, the component is defined as follows:

Definition 2 (Component): Component in user-customized configuration design pattern is part or component of mechanical product that encapsulates internal structure, and can enter into assembly phase directly on the premise of meeting interface criteria in certain field.

The definition could be understood from following points:

1) Component is a physical entity, component consumers could assemble components into products through a simple way similar to "building blocks", and the product composed of components could add, delete or replace some components according to needs.

2) Component owns standard interfaces and could be used by different products.

Component interface and structure is divided. It exchanges energy, matter and information with the outside world only by interface.

3) The premise behind assembling product by components is that the component is available. The standardization of component interface determines the possibility of mass production, so the component can be circulated in the market at a lower price and users can obtain components easily.

The definition of component is given to increase the diversity and convenience of the user optional components. The industrial robot of manufacturing companies has been realized reconfigurable modular structure, but due to the upper standard not unified and technology blockade in companies, industrial robots can only achieve modules reuse within the same company, and cannot achieve matching across the companies. This factor limits the user's configuration design. The component provides a standard interface, and encapsules component structure which the companies are highly sensitive to but customer is not concerned about. Companies can implement the same interface by different techniques. In this way, companies can release components on the premise of maintaining core technology. Component encapsulation also blocks technical view for the users, which makes it easy for the users to configure product.

It's important to note that the concept of module and component can be easily confused. In the research of mechatronics product modularization, a module refers to the physical realization of a function [129]. The module focus on dividing the components of the product into several parts (which is called module) according to the dependencies between components, and each module has an independent function. For example, in reconfigurable modular robot, a module usually contains processors, motors, transmission devices, etc., and the module can be tested and run independently [130][131]. In the research of user-customized configuration design pattern, a single component doesn't necessarily realize one independent function. It focuses on encapsulating part or component of mechanical product, and the interaction between components is achieved through standardized interfaces only.

In fact, the concept of module and component are not completely different. With the continuous development of product modularization research, the module interface is also required to be standardized, which is identical to the requirement on

component. For highly modular product, a module of the product can become a plug-and-play component by being encapsulated. On the other hand, the idea of modularization can be introduced into the design of component. The components having more interaction could be designed as a module, and then encapsulate this module to a big component. Doing this could reduce the workload of configuration design.

It can be known from definition 2 that component is a kind of physical entity, and it exists in realistic physical space. But the user-customized configuration design activity is not assemble the actual physical components, but for virtual assembly in the computer. Components, therefore, need to be described abstractly for computer to identify. The service component is used to describe physical component, and its specific definition is as follows:

Definition 3 (Service-component, SC): Service-component is the abstract description of component using formal language based on specifications.

The connotation of service-component includes the following aspects:

1) The SC is a conceptual model that can be stored in the computer, and it is the abstract description of physical components in the real world.
2) SC description a component framework rather than a specific component, which only have properties but no properties values. It needs to be assigned to describe specific component.
3) In service-oriented architecture of software development, the software components are encapsulated as services, and software developers could complete their business by invoking services rather than using component directly. Similarly, users could complete virtual product assembly by invoking SC in the computer rather than using physical components directly in the real world.

The third concept needs to be clarified is user. In user-customized configuration design pattern, the user is defined as follows:

Definition 4 (User): User refers to all people that can make design activities. There are two kinds of user: the enterprise or individual that use the product directly (i.e. customer); the enterprise or individual that provides product solution for product end-user.

The fourth concept needs to be clarified is user-customized design. Its definition is as follows:

Definition 5 (User-customized configuration design): Under certain constraint conditions, user assembles SC selected from the SC library, and then obtains product design results satisfying his/her own needs.

By user-customized configuration design, a configurable product is obtained. In configuration design field, there are a lot of definitions about configurable product. For example, a product is configurable when each individual product is adapted to the needs of a particular customer order [132]. A configurable product is also defined as a product which is composed according to the demands of a particular customer order based on a generic product structure [133].The main feature of a configurable product is its modular structure, and the possibility to create individual products by selecting different predefined components [134]. In design process, the configurable product is generally composed by common components and special components [135].For industrial robot, common components are the components required in different products, such as the wrist, the drive system, etc. Special components are used to achieve a specific function for different users, such as the various sensors, end-effectors, etc. In industrial robots design, a design template can be built by distinguishing different component types, which can improve the configuration efficiency. Based on a comprehensive collection of existing research, the concept of configurable product is defined as follows:

Definition 6(Configurable Product): Use $ComCpt_i$ to represent common components, and $SpeCpt_i$ to represent special components. Set $\sum_{i=1}^{m} ComCpt_i$ as the collection of $ComCpt_i$, and $\sum_{i=1}^{n} SpeCpt_i$ as the collection of $SpeCpt_i$. Product is represented by P, and

$$P = C\{\sum_{i=1}^{x} ComCpt_i, \sum_{i=1}^{y} SpeCpt_i\}.$$

Wherein, x, y, m, n is integer,

$$\sum_{i=1}^{x} ComCpt_i \subseteq \sum_{i=1}^{m} ComCpt_i, \sum_{i=1}^{y} SpeCpt_i \subseteq \sum_{i=1}^{n} SpeCpt_i.$$

C refers to constraints. The constraints here refer to the rules must to be followed to guaranty the product feasibility when assembling components to product. For example, when one gear meshes with another, the required transmission ratio determines the number of gear teeth; when a sensor is connected with a controller, not only the interfaces but also the communication protocol need to be consistent, etc.

The last concept needs to be clarified is design platform. A platform is usually used to achieve a synergy of sales force automation, product design, manufacturing planning, and supply chain management within a consistent framework[136]. It provides information sharing and communication mechanism for all the participants in user-customized configuration design pattern. The definition is as follows:

Definition 7 (Design Platform): The design platform is formed of hardware and software environments and technical processes supporting user-customized configuration design pattern realization.

The platform realizes user-customized configuration design pattern, so it must be open. The openness is embodied in three aspects: the first is that people using the platform are diversified, so it must be able to be used by all stakeholders such as users, component manufacturers, designers, etc.; the second is that the enterprises publishing SC on the platform and the enterprises using SC through the platform are diversified, so the platform must provide component release criterion, and components meeting the criterion can enter the platform as SC. The SC publisher could use SC as SC user, and the SC user also could SC publisher to publish SC on the platform; the third is that the user can not only design product by their own, but also can choose design results finished by other users, so the platform must support all users design product and share design result.

On the open design platform, the more basic components could be invoked by more variety of industrial robot, but also increase the configuration complexity for the user. As a result, the component designer should consider the technical issue of component granularity. The component designer should find a balance between increasing component sharing and decreasing configuration complexity, and design the component according to the interface criterion of the platform. In the short term, these requirements increase component development cost. But in the long term, the

components with appropriate granularity will be called more frequently after user-customized configuration design pattern is received by people, as a result component develop companies can get more benefits. These benefits are the driving forces for the component develop companies to improve component design continuously.

2.3 Business Mode Analysis

2.3.1 User-customized configuration design business mode

From the definition 1 we know that user-customized configuration design pattern is a guidance method. The guidance method takes design method changing as the core, and the changes of development approach bring the role changes of relevant enterprises such as industrial robots manufacturers, parts providers and developers in industry chain. As stated in the introduction, the existing business mode can't adapt to user-customized configuration design pattern completely, and it is necessary to research a new business mode to ensure the smooth progress of this design pattern. The business mode of user-customized configuration design pattern is shown in figure 2-2. Each participant's activity and its interactions with other participants in the mode are as follows:

①Customers design product on the platform, or purchase product designed by other designers. Then pay online and create order;

②Customers publish the requirement that can't be met by user-customized configuration design or purchase on the platform, and other designers could view these requirements and propose solutions;

③ Designers design component or product according to the published requirements and then sell the design result on the platform;

④Component manufacturers accept order online, also could purchase the design result designed by designer to manufacture;

⑤Component manufacturers deliver the component purchased by customer to product provider through offline distribution. The product provider is also the platform creator. It is responsible for establishing experience store in various areas and providing after-sales service for the product designed in user-customized configuration design method. The product provider assembles component to product according to customer's order in the experience store, guides customer on using the

product, and then deliver the product to product logistics provider;

⑥The product logistics provider selects appropriate logistics project for the product and then distributes the product to customer;

⑦ Quality supervision departments supervise the various products and service quality of the participants, and ensure that the entire process and the final products meet national standards. Platform service provider provides technical support for platform running.

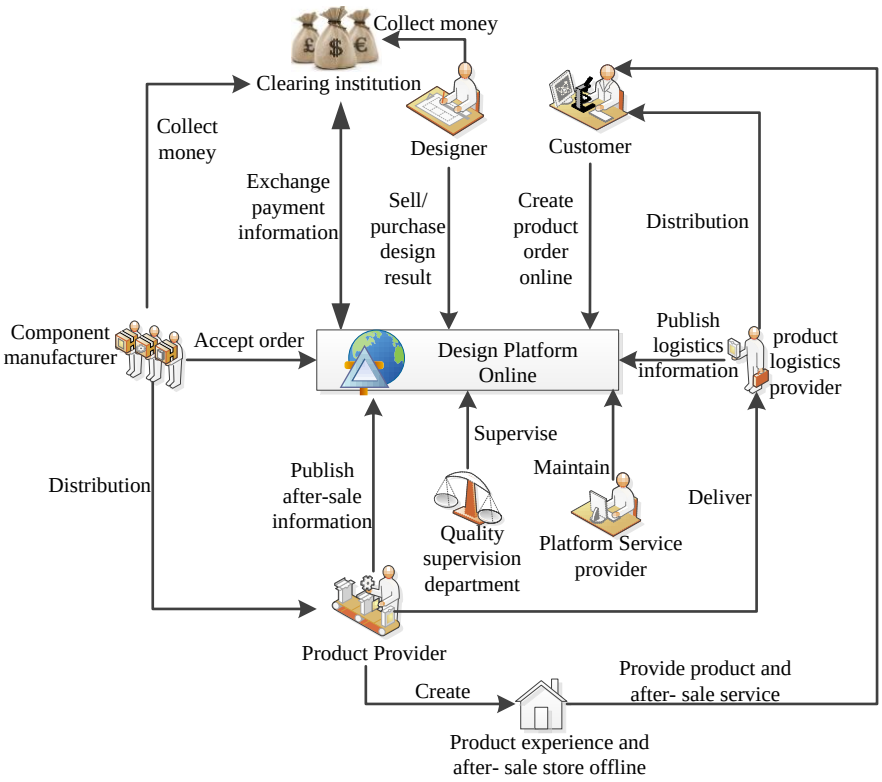


Figure 2-2 User-customized configuration design business mode

In the whole process, the first four steps are online activities, and the fifth step is offline activity. The trading activities taken by designer on the platform belong to the C2C mode. Any individual with the ability to design can sell their own design results through the platform, and also can purchase other designer's design results. Customer purchases components online, and accepts components distribution offline. This activity belongs to the B2B or B2C mode-- B2C when the customer is individual and B2B when the customer is enterprise. Customer creates product order online and experiences product and services in offline store, this activity belongs to the O2O mode. Therefore, user-customized configuration design business mode is a new mode with a combination of all kinds of existing business modes. This mode unifies all

participants in the supply chain, and manages the logistics, cash flow, information flow effectively.

Design platform is the core participant of the business mode. It doesn't provide product manufacturing service, but it provides customized product services, experience product service, assemble product service and after-sale service. These services provide customer a lot of additional services while purchasing product. Also, the platform allows designers to trade their design results, which make the designer obtain economic benefits.

Due to provide various services for customers, the platform can gain numerous customers visits, and this is the motivation for component manufacturers and designers to register on the platform. The activity of each participant centers around user-customized configuration design. The design method provided by platform and the whole process depended by the design method is unique to the platform. Other enterprises can imitate an information system, but they can't replicate the user-customized configuration design method. So this business mode has non-repeatability and strong competitiveness in the market.

2.3.2 Key technologies of user-customized configuration design business mode

Through the figure 2-2, it can be found that all participants' activities online must depend on the design platform, that is, the realization of the design platform is the core of carrying out the business mode smoothly. In the process of realization there are the following several key technologies to be solved:

The representation of robot modules is the basic for configuration [137][138], so the unified expression method of the industrial robot component needs to be determined firstly. The necessity of components representation embodies two aspects: firstly, industrial robot components involve different fields, including mechanical components, electrical components, etc., and it does not only consist of a physical connection between the various components. The connection involves electrical parts, control signals, etc. In order to store all of the components on the same platform, the unified expression method of the multidisciplinary component needs to be determined; secondly, the platform has openness. All potential users having design capabilities can act as designers and publish their own design results on the platform, so there will be a very large number of components on the platform. To platform can accurately call the released components and component providers can quickly search various

components, some criterion need to be satisfied with when the components being published.

Secondly, a user-customized template for industrial robot needs to be developed. The platform supports user-customized configuration design, but users are not professional designers, and hence cannot guarantee the manufacturability and rational construction of the product. Therefore, on the base of components expression, the platform needs to provide a customized template to guide the user in the design activities. A customized template is defined as follows:

Definition 7 (Customized configuration design template): The template is a generic and abstract product structure composed by SC. User instantiates the structure via assigning to the SC, then products model with specific features can be obtained.

The connotation of customized configuration design template includes the following aspects:

1) The external view of the template is similar to actual product, and it is a simple view on behalf of the general product structure, which can be understood quickly by the user;
2) Inside the template it contains mutual constraints between service components. Each part of the final product model obtained by the template meets the constraints, which ensures the feasibility of the configuration result;
3) The template provides the performance calculation of the final configuration result, including kinematics, dynamics, vibration, accuracy, etc. The calculation results are visually represented to the users;
4) The final product model consists of instantiated service components. These components have a one-to-one correspondence to physical components, and a practical physical product can be assembled through contrasting product model. The physical product is in conformity with the data provided by the template in terms of performance.

Thirdly, the user-customized configuration design process needs to be analyzed. In the process, the configuration orientation of each step must be determined by the user. At the same time, the user's operations should be as simple as possible and leave most of the work to platform. As the platform is responsible for providing a variety of functions such as the SC instantiation method, instantiation order, component retrieval

method, parameter calculation, etc., it is necessary to study on various algorithm of the platform.

2.4 Validation of Proposed User-Customized Configuration Design Pattern through Simulation

Based on the configuration design theory and SOA idea of software development, the thesis puts forward user-customized configuration design pattern. In software development, the software component can be obtained by running the completed programming. Compared with traditional software development method, the advantage of this architecture can be reflected intuitively. But as shown in figure 2-2, after design activity finished, the industrial robot components need to be manufactured, transited, etc., and the business mode running is not as simple as the software. So the advancement of the business mode remains to be further validation. In this section, system dynamics models for the user-customized configuration design business mode and traditional business mode are built to analyze their design and production process. System dynamics is a modeling and simulation method that is widely used in production process, and it is very suitable for the simulation of complex dynamic systems [139]. The performance advantage of the proposed model is validated by simulation analysis, which provides rational basis for realizing product design platform next step.

2.4.1 Analysis of traditional business mode and user-customized configuration design business mode

As a kind of mechatronics product, industrial robot can achieve a certain degree of configuration design at present, that is, some common features can be realized by reusing previous design results, but on the whole the product still needs to redesign according to the needs of customers. So the entire process of traditional business mode belongs to engineering-to-order. The company designs products according to users' requirements, generates orders, and organizes production processes based on the orders. The operation process of the entire traditional design and production process of industrial robot is shown in Figure 2-3. Users put forward their demands, the design department of the company designs several products to meet the needs of each user, and then the production department arranges the production plan. The entire production process begins with basic components manufacturer. It produces components and parts in accordance with orders, and promotes the entire production

process to downstream until the products are submitted to users.

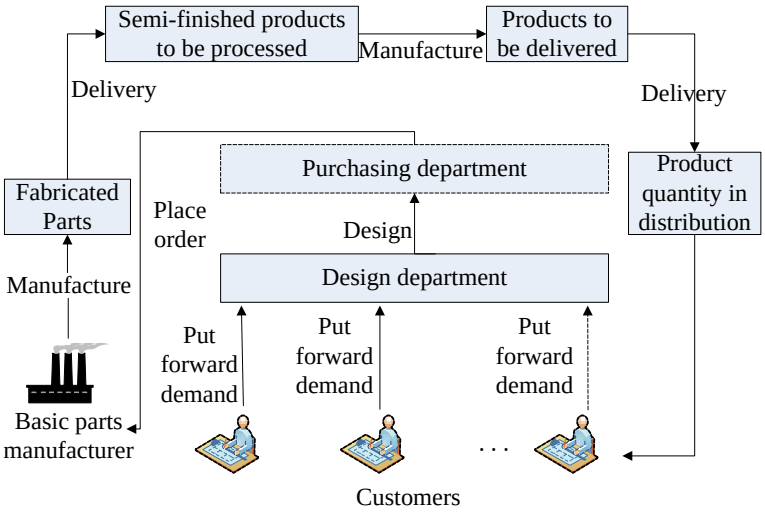


Figure 2-3 Traditional business process of industrial robot

In user-customized configuration design pattern, the production mode is assemble-to-order. The platform is the core of the whole process. Design, manufacture, assembly and other departments work in cooperation. The running process is shown in Figure 2-4.

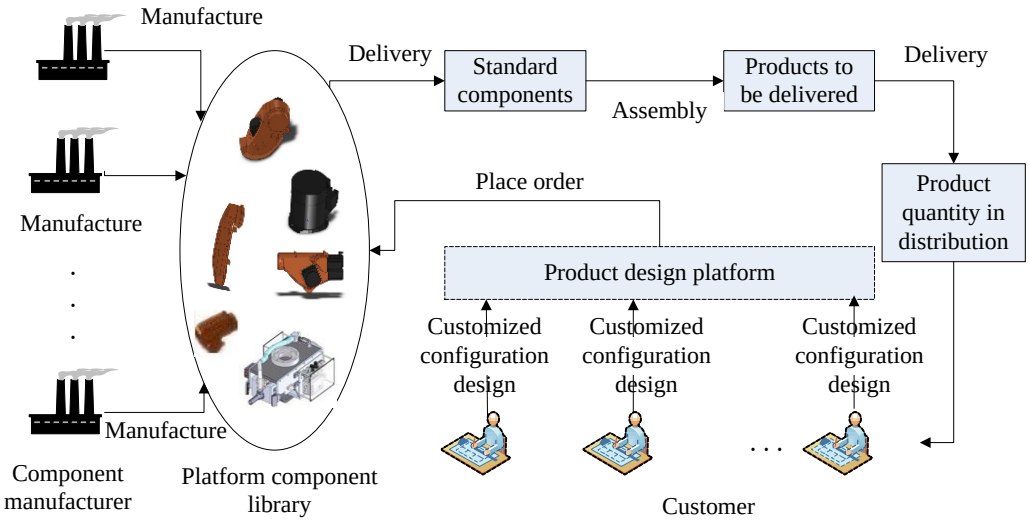


Figure 2-4 User-customized configuration design process of industrial robot

The user selects service components from the service component library according to the clarified needs, configures the product on the customized configuration design template, and sends his/her order to the component manufacturer after the completion of configuration design. The product provider assembles the components to product based on the user's order. Various component manufacturers

publish their components on the platform, and then a component library is formed, in which the components can be selected freely. The core of the whole process is the assembly stage. Product provider purchases quantitative standard components in advance as inventory, hence the company can arrange for the assembly activities immediately when they receive customer orders. At the same time, the company continues to purchase components based on the number of orders and inventory.

2.4.2 System dynamics model

Chen Guohua and his research team proposed a system dynamics simulation method of different supply chain mode [140]. But they focus on delivery ratio research and did not consider the design pattern influence on order. Based on their research and the analysis of the entire design and production process in section 2.4.1, the simulation models of the two business modes are built using system dynamics on Anylogic simulation platform. In system dynamics model, "stock" is used to describe things increasing or decreasing over time in the real world. In the process of things increasing or decreasing, the "flow" generated, and the difference between inflow and outflow of the stock is the stock value. "Parameter" represents a constant or variable outside the model which having impact on the model running. "Auxiliary" represents the internal information of model, and it describes the relationship between variables.

The system dynamics model of traditional business mode is shown as Figure 2-5. The entire process contains two parts: order processing and product production.

On the step of order processing, new demands emerge continually in market. It takes time to design products according to demand, so the stock "Demand accumulation (DA)"generates in the case of the demand not being met immediately. The value of DA is decided by the difference between its inflow demand generate rate (DGR) and outflow order generate rate from demand (OGRFD). The company started to design products after receiving users’ demands, transfer product requirements into orders, and then began to arrange production according to orders. In this process, the order generate rate from demand is determined by "Demand accumulation" / "Design time (DT)".Order accumulation (OA) is determined by the difference between its inflow order generate rate from demand and outflow order finished rate(OFR).The order accumulation minus the orders have been processed (including the orders being processed-"Processing semi-finished products inventory(PSFPI)"and the orders have been completed - "Finished products inventory(FPI)") is the required semi-finished

products order quantity (SFPOQ).

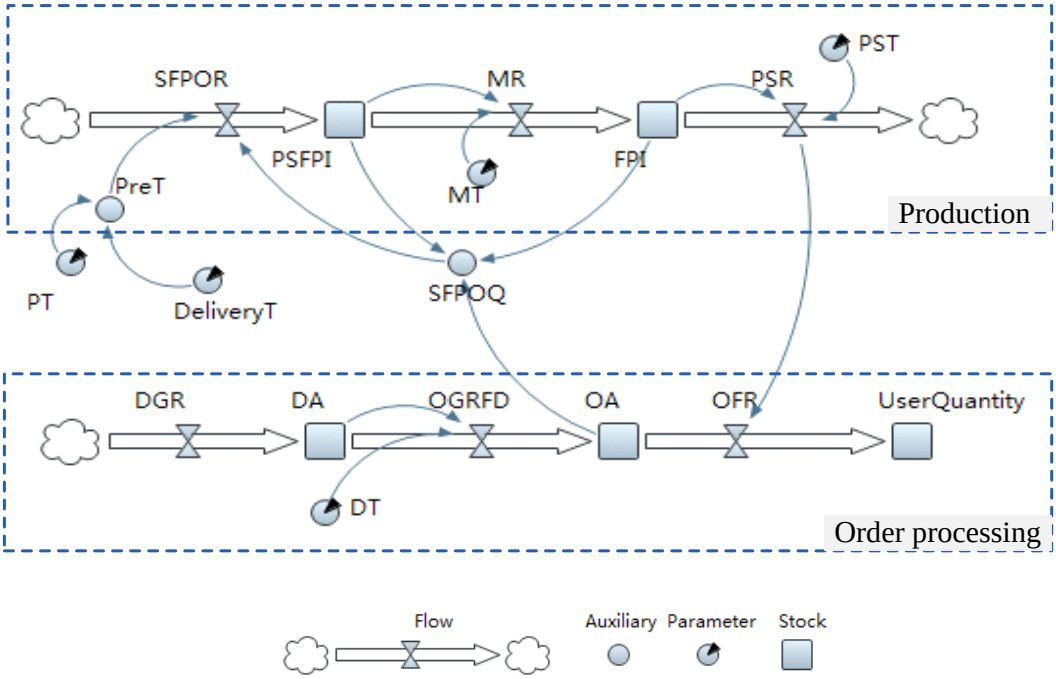


Figure 2-5 Traditional business mode

On the step of production, semi-finished products need to be purchased after receiving orders from design department, so there is no semi-finished products inventory at first. It takes some time to become products from semi-finished products, so the stock "Processing semi-finished products inventory (PSFPI) " generated in production process. Similarly, the stock "Finished products inventory (FPI)"generated because it is not possible to delivery all the finished products immediately. On this step, the stock "Processing semi-finished products inventory" is determined by the difference between its inflow semi-finished products order rate (SFPOR) and outflow manufacturing rate(MR). Similarly, the stock "Finished products inventory" is determined by the difference between its inflow manufacturing rate and outflow product shipping rate (PSR). The product order is finished when the product begins to be shipped, so the product shipping rate is equivalent to the order finished rate.

The system dynamics model of user-customized configuration design pattern is shown as Figure 2-6. The process also contains two parts: order processing and product production.

On the step of order processing, this pattern also generates stock "Order accumulation (OA)", which is similar to the traditional mode. The accumulated orders minus the finished products order (FPI) is the component quantity to be assembled

(CQA). Notice, however, that the users design the orders by themselves, and the design time is the time required for each user in order to design their own products. It is different from the first pattern, whose design time is the time needed by the design department in order to design all orders. Assume that every user's design time is the same, so the demand generation rate (DGR) moves backward a length of design time on the timeline is the order generate rate (OGR).

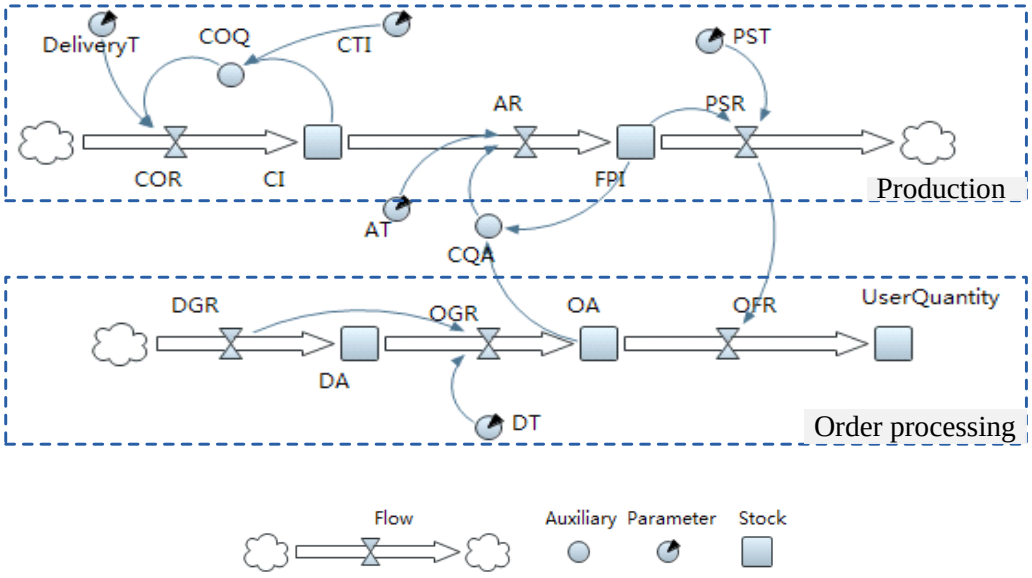


Figure 2-6 User-customized configuration design business mode

On the step of production, since the components used in design process are selected in the component library, company can purchase a certain amount of components from the component provider in advance as target inventory. On the premise of maintaining normal production, the inventory is as little as possible. In this case, company can immediately plan for the assembly using the inventory of components when the user orders arrive. When the component inventory is less than the target inventory, the components are ordered in time, and the component order quantity (COQ) is determined by the difference between component target inventory (CTI) and component target inventory (CI). The stock "Component inventory (CI)" is determined by the difference between its inflow component order rate (COR) and outflow assembly rate (AR).

2.4.3 Simulation results and analysis

Before simulation, the model parameter is set. The simulation time unit is set in day, the simulation step is set in 0.2, and the simulation length is set in 80. The generation rate of demand in line with a normal distribution whose mean is 10 and

variance is 10.The random number distribution area of the normal distribution is [0,20].At the thirtieth day, the generation rate of demand is changed. Its mean is changed from 10 to 15. When the user demand generation rate increases, the components target inventory in the second pattern is modified into 35 by assembly department in order to accommodate the increase of demand, while the assembly time and shipping time is modified to 3 days. Similarly, production time and shipping time will be adjusted to two days in the first pattern when the user demand generation rate increases.

The model parameter setting and variable formula in Figure 2-5 are as follows:

Table 1 Parameter settings for traditional design model

Parameter	Initial value	Parameter	Initial value
Production time (PT)	3	Processing semi-finished products inventory (PSFPI)	0
Delivery time (DeliveryT)	3	Finished products inventory (FPI)	0
Manufacturing time (MT)	3	Demand accumulation (DA)	0
Product Shipping time (PST)	3	Order accumulation (OA)	0
Design time (DT)	3	Demand generate rate (DGR)	normal distribution(10,10)

Table 2 Flow settings for traditional design model

Flow	Formula
Semi-finished products order rate (SFPOR)	Semi-finished products order quantity / Preparation time
Manufacturing rate (MR)	Processing semi-finished products inventory / Manufacturing time
Product shipping rate (PSR)	Finished products inventory / Product Shipping time
Preparation time (PreT)	Production time + Delivery time
Semi-finished products order quantity (SFPOQ)	Order accumulation –Processing semi-finished products inventory –Finished products inventory
Order generate rate from demand (OGRFD)	Demand accumulation / Design time
Order finished rate (OFR)	Product shipping rate

For comparison of the overall performance of the two systems, the initial value of time in Figure 2-6 is set to the same value as in Figure 2-5. The specific settings are as follows:

Table 3 Parameter settings for user-customized configuration design model

Parameter	Initial value	Parameter	Initial value
Delivery time (DeliveryT)	3	Component inventory (CI)	30
Component target inventory (CTI)	30	Finished products inventory (FPI)	0
Assembly time (AT)	3	Demand accumulation (DA)	0
Product Shipping time (PST)	3	Order accumulation (OA)	0
Design time (DT)	3	Demand generate rate (DGR)	normal distribution(10,10)

Table 4Flow settings for user-customized configuration design model

Flow	Formula
Component order rate (COR)	Component order quantity / Preparation time
Assembly rate (AR)	Component Quantity to be assembled / Assembly time
Product shipping rate (PSR)	Finished products inventory /Product Shipping time
Component order quantity (COQ)	Components target inventory - Component inventory
Order generate rate (OGR)	Delay(Demand generate rate, Design time)
Order finished rate (OFR)	Product shipping rate
Component Quantity to be assembled (CQA)	Order accumulation - Finished products inventory

The resulting order generation rates of the two patterns are shown in Figure2-7. Simulation results for production inventory and orders accumulation are shown in Figure 2-8.

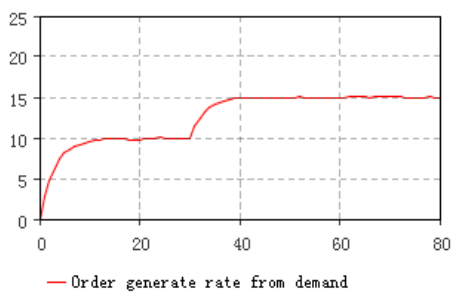


Figure 2-7a Order generate rate from demand in traditional design pattern

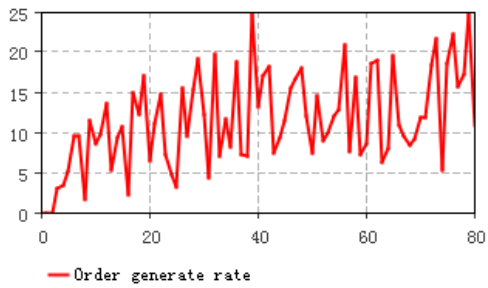


Figure 2-7b Order generate rate in user-customized configuration design pattern

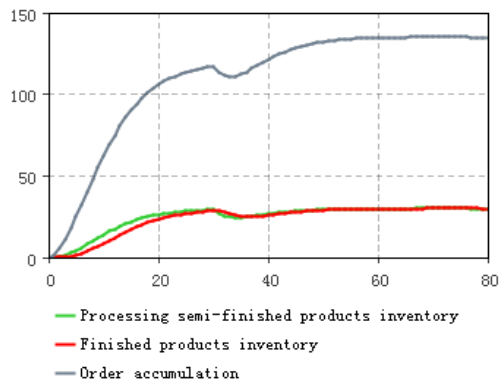


Figure 2-8a Simulation results of traditional design pattern

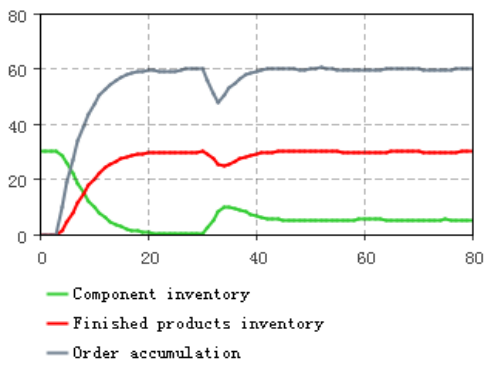


Figure 2-8b Simulation results of user-customized configuration design pattern

The simulation results are based on the following assumptions:

① The time required to design a product in figure 2-3 and 2-4 is the same. Because the models only simulate the whole process from demand generation to product delivery in the two modes, and don't simulate the specific product design process and design time of the two modes.

② When designing the same product, the performance of product designed through the two modes is the same.

③ In figure 2-4, the components in component library can fully meet customers' demand.

The companies organize production after receiving orders in these two patterns, thus order backlog will appear as shown in simulation results. In the case of the same user demand generated rate, order backlog of traditional pattern is more than 100, but the order backlog of the new pattern is only about 60. This result means that the response speed to user demand in user-customized design configuration pattern is much faster than in the traditional pattern. In addition, the inventory level directly determines the production cost of the products. From Figure 8a and 8b, the total amount of processing semi-finished products and finished products inventory achieves stability at about 50 in traditional production pattern, while the total amount of component and finished products inventory achieves stability at about 30 in user-customized configuration design pattern. The result shows that the latter pattern

ensures faster response speed to user needs in low inventory level.

When user demand mean is changed from 10 to 15, the inventories of these two patterns do not change significantly after the system reaches a steady state. But in Figure 2-8a, the order backlog increases about 10, and in Figure 2-8b the order backlog does not change significantly. The result shows that the adaptability of the latter pattern is stronger when user demand is changed. In traditional design pattern, even if the company adjusts production capacity, the orders accumulation will still increase due to the transportation time constraints of semi-finished products.

In addition, the design time of the two patterns are set to 3, but Figures3and 4show that in traditional design pattern, the product is designed by the company, and in user-customized configuration design pattern, the product is designed by the user, which means there are as many designers as there are users. Therefore, from the perspective of the entire market, the actual design time of the new pattern is shorter than the traditional pattern, so as to shorten the time from demand generation to final product sent to users.

In the initial set up stage, the types of industrial robots designed by users on the platform are very few, and some components are needed to be redesigned by professional designers to meet most demands. But with the continuous development of the platform, as many more people are increasingly using the platform to design and purchase product, more and more components development companies or individual will publish their components on the platform for users calling to obtain benefits. When the number of components published to the component library is large enough, the majority of user demands can be meet through the platform by user-customized configuration design. Then the advantages in terms of inventory cost and user order response quality will be reflected and the companies and users will achieve a win-win results.

2.5 Chapter Summary

According to the characteristics of industrial robots, user-customized configuration design pattern is presented to achieve rapid development of industrial robotics. Firstly the theory source of user-customized configuration design pattern is introduced, and then the related concepts are expounded. Secondly, the corresponding business mode of user-customized configuration design pattern is given, and the key technologies to realize business mode is studied. Finally, system dynamics models

were established for user-customized configuration design business mode and for traditional business mode of industrial robots by Anylogic simulation software. The simulation analysis proved that the proposed design pattern has obvious advantages in cost and user order response than traditional mode.

CHAPTER 3 COMPONENT-BASED MODELING FOR PRODUCT PARTS

Product is constituted by parts. In user-customized configuration design pattern, the parts meeting certain interface standard are called component. The real-world physical components could be stored in the computer after formal description, and this is the basis for user-customized configuration design. The description method depends on the intended use. Bai Jing uses matter-element model to desicribe product module in the design process of CNC system configuration [141]. This model is mainly focused on the use of the module in the configuration process. The proposed user-customized configuration design pattern in this thesis is implemented on an open platform. There will be more and more of the components in the platform component library. How to efficiently and accurately find the needed components is a problem to be considered. Ontology-based product representation is to solve the problem of efficient retrieval [142][143]. But in the existing literature, the ontology representation can only solve the problem of efficient retrieval, but can not provide support for the use of components. Based on the above analysis, this chapter build the component-based modeling of product parts, and provide support for the retrieval and use of the product components.

3.1 Component-based theory and method

3.1.1 Formal description method of things

Formal methods are mathematically based languages, techniques, and tools for specifying and verifying complex systems [144]. Formal methods are commonly used including algebraic method, the state machine, Petri nets,etc.. However, a common drawback of these methods is that they can’t be corresponded directly to the real world, so the the transition from the real world to the formal model is difficult. Object-oriented thinking provides a description approach which corresponding real world. The method view things in the real world as objects. To be able to carry out a computer simulation of the real world, the things in real world are abstracted to the objects in computer. However, the number of things in the real world is very much. Representing every thing in the computer is unrealistic, and is not necessary. So the objects having the same attributes and behavior are packaged to class. For example, in the real world there are a lot of people. when described in the computer, the people can be abstracted as a *Person* class. In the specific use, the class is instantiated to a specific person (such as Jack) by value assignment for this class. This person has all attributes and behavior of the *Person* class. Figure 3-1 illustrates the object-oriented thinking by an

planting example.

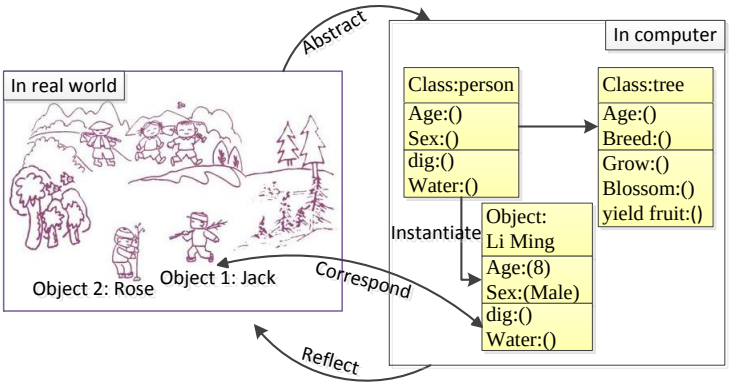


Figure 3-1 Object-oriented thinking

In the computer, object-oriented thinking builds links and interactions between the various classes in a way closing to human thinking, and ultimately reflects the real world. The implementation method can be hiden by encapsulating, the class can be extended by inheritance, and different execution results with the same object can be achieved by polymorphism, etc.. These characteristics make the system easy to be maintained and and extended.

3.1.2 Ontology and domain of discussion

Ontology is a philosophical concept. It researches the nature of things existing. In the computer field, ontology is an explicit formal specification of conceptualization [145]. Domain of discussion is a collection of entities as research objects in particular area. For example in the study of industrial robots, the domain of discussion is all the parts that constituting industrial robots. Setting up the research domain of discussion is *D*, the conceptualization of *D* is represented as a triplet structure <D, W, R>. Where *W* is the largest state of things in *D* (or the largest possibly world) collection, *R* is the collection of concept relationships in domain space [146]. The conceptualization is defined clearly, and represented formly which can be understood by the computer, then the ontology is constituted. The ontology cantains five basic modeling elements: cIasses, reIations, functions, axioms and instances. Usually classes is written to concepts [147]. The concept can represent anything, the relationship represents the interaction between the concepts, the function is a special relationship, the axiom is on behalf of true statement, instance is on behalf of the element, i.e. the object.

Ontology describes domain of discussion through the five modeling elements, and establishes links between the computer and the domain, so that the computer can understand the domain of discuss to be studied from the semantic level. Because of the relationship

between concepts, the computer can perform semantic reasoning, which is the reason that the use of ontology representation method can improve the recall ratio and precision ratio of concepts.

The ontology and object-oriented thinking both have the concept of class and object, but the use purposes of the two concepts is different. Object-oriented thinking is applied to the realization process of specific system such as software system or a certain thing. The description of the class is focus on the class operation achieved in the system. It is a functional level of description. The description of semantics in the object-oriented method is based on natural language, and there is no precise definition. For example, in a system there is a class *A*. For *A*, it can have different semantic interpretation, and the different semantics don't affect the achievement of the system. The ontology describes the common knowledge in domain from the semantic level. It does not focus on how a specific class playing a role in the system, and describes the location of the class and the interaction between classes in the domain of discuss from the knowledge level. These descriptions can be seen as domain knowledge to achieve a specific system in future. The objects and classes in object-oriented thinking are at different levels. The object will not appear in the class diagram, and the object diagram is a description of the system state at a certain moment. But the object in ontology is an important part of the ontology, and it is a concept with the smallest size [148]. Class can be seen as a collection of objects, they exist in the same body.

3.1.3 Componentization and servitization

From the definition 2 of Chapter 2, it can be known that the most critical issue for changing the product parts into components is the component interface. When the parts are released to the platform, no matter how the internal structure of the part, as long as the interface meets certain standards, and can be replaced with other part having similar functions, it can be said that the part is in line with the componentization standard. In user-customized configuration design pattern, the interface standard is defined and released by the platform. This process of unified specification and definition for the product parts' interface is called componentization. After componentization, the product parts become components with standard interface.

From the figure 2-5 in Chapter 2, it can be seen that in the user-customized configuration design pattern, users design product on the network design platform, rather than assemble the physical components in the production site. Therefore, the premise of the design is to formalize the component, and release the component to the platform to provide users with

configuration services. This process is called the servitization for component. After servitization, the components become service-component, and be stored on the platform.

3.2 Component-based modeling for product parts

3.2.1 The object model of component

The parts will become components after the interfaces of parts being standardized. According to object-oriented representation, the component could be described from attribute and operation. Simultaneously according to the definition of component, the component encapsulates the internal structure of parts, and interacts with outside only by interface. So the interface is also an important parameter to be described. The object layer of service-component could be described as follows:

$$Ob = \{Atr, Int, Ope, Inc\} \tag{3.1}$$

Atr is the attribute set of *S*. $Atr=\{a_1,a_2,\ldots a_n\}$, wherein a_i is the component technical feature such as power, size, inertia, weight, etc. The domain of value of *Atr* is represented to $Atr^v=\{ a_1^v, a_2^v,\ldots a_n^v \}$, and the special value of *Atr* is represented to $Vatr=\{ Va_1, Va_2,\ldots Va_n \}$, similarly hereinafter.

Int represents the interface of *S*, and $Int=\{i_1,i_2,\ldots i_m\}$. The i_m describes component information interacting with outside such as the physical interface type, physical interface attribute and physical flow through the interface. The physical interface type contains mechanical contacting type such as kinematic pair; data transmission type such as interface connected by data cable; non-contact type such as infrared data interface. The physical interface attribute contains mechanical interface size, interface communication protocol, etc.. The physical flow contains signals, data, energy, physical entity, etc.. When there is a connection relationship between two components, there must be two matching interfaces on the two components. The meaning of matching includes the same interfaces type, the same interface attribute, and the same physical flow. The interface is a special attribute, and all the invocations for the service-component must via the interface;

Ope represents the operation of service-component, and $Ope= \{o_1,o_2,\ldots o_j\}$. The o_j describes the function/service provided or relied by the component;

Inc is the constraint relations of the service-component's attribute, and $Inc= \{c_1,c_2,\ldots c_k\}$, wherein c_k describes the impact on other attributes when one attribute inside the service-component changing. *Inc* can be represented by a set of constraint equations or production rules. In the constraint of *Inc*, the attribute or attribute value could be changed,

which makes the service-component could be extended within a certain range. When the *Inc* value is empty, the service-component is a standard component and can't be extended.

3.2.2 The business model of component

The object model of component describes the technical parameters. while the business model of component describes the component from a management point of view, and it can be expressed as follows:

$Ma=\{m_1,m_2,\ldots m_n\}$, where m_i represents the information needed by the platform management such as service-component version, creation time, the publisher, frequency of use, manufacturers etc., and the information issued by the manufacturer such as component suppliers price, origin, ease of use, adequacy of supply, delivery, etc.. *Ma* also belong to the data attribute of ontology concepts, but it does not describe the essential properties of the component. It is the information generated and used in component publishing, manufacturing and using. For a platform integrating stakeholders, *Ma* is essential information for platform running and maintenance.

3.3 The servitization of product component

3.3.1 Representation based on ontology

Component-based description model of the product parts make the user can use component in specific product configuration design. However, it is needed to release the component model to the platform as service-component according to certain specification, so that user can find the needed components, and the component can truly provide service for user. In this chapter, ontology-based representation is used to unify the concepts of the component-based description model. Record the domain of discourse to *D*, and the ontology description of *D* is *O*, there is:

$$O=<C,R,A,I> \tag{3.2}$$

$C=\{c_1,c_2,\ldots.c_n\}$, wherein *C* is the concept set of *O*. The concepts contain general concepts suc as torque changing, support, etc., and also contain concepts representing *S* such as base, servo motor, etc. There is formula $C\subseteq D$.

$R=\{r_1,r_2,\ldots r_m\}$, wherein *R* is a binary relation set between concepts, and there is formula $R\subseteq C\times C$. $r_i(c_x,c_y)$ means that there is r_i relation between the concept c_x and c_y. In this relation, the domain of definition of r_i is c_x, the domain of value of r_i is c_y. The relation has direction, for example, $r_i\ (c_x,c_y)$ is different from $r_i\ (c_y, c_x)$, and $r_i\ (c_x,c_y)= -r_i\ (c_y, c_x)$;

$A=\{a_1,a_2,....a_n\}$, wherein A is an axiom set of O.

$I=\{i_1,i_2,....i_n\}$, wherein I is an instance set of O.

3.3.2 Description method for service-component

The service-component is recorded to S, there is:

$$S=<ID,On,Ob,Ma> \tag{3.3}$$

Wherein, ID is a unique identification for S, On is the ontology layer description for S, Ob is the object layer description for S, and Ma is the business layer description for S.

According to ontology representation method, there is:

$$On=\{c_i,R^{ci}\} \tag{3.4}$$

$R^{ci}\subseteq R$, and it is the relation set between S and other concepts in D. $R^{ci}=\{r^{ci}_1,r^{ci}_2,...r^{ci}_n\}$, wherein r^{ci}_i means that there must be a concept c_j having a r^{ci}_i relation with concept c_i.

In ontology theory, the binary relation between concepts could be seen as a special attribute called relation attribute. The value of relation attribute is the concept. For example, for the concept c_i, there are relations $r^{ci}_x(c_i,c_m)$, $r^{ci}_x(c_i,c_n)$ in the ontology, so the concept c_i has relation attribute r^{ci}_x, and the domain of value of this relation attribute is $\{c_m, c_n\}$.

In addition to relation attribute, the ontology concept also has data attribute. The object layer of service-component represents the relation attribute. Data attribute is the natural character of the concept itself. For example, the Ob and Ma of S is the data attribute, and the value range of data attribute is the parameter value of Ob and Ma.

The two types of attribute is shown in Figure 3-2. In the figure, concept c_x has relation attribute r_i, data attribute a_i. The value of r_i is c_y which means there is r_i relation between c_x and c_y, and the value range of a_i is a_i^v。

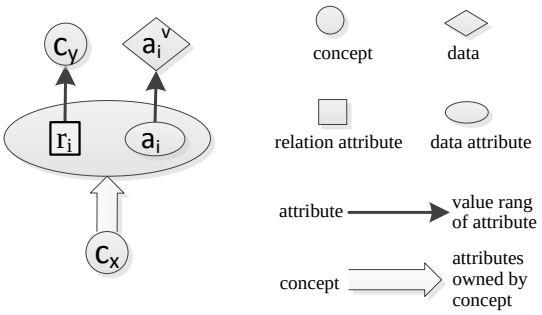


Figure 3-2 Two types of attribute of concept

The service-component is presented as the concept of ontology, and then the relationship

between the concepts are added. So that the domain knowledge reuse is realized. Service-component describes the component from three layers of ontology, object, business. The role of the three layers in user-customized configuration design is shown in Figure 3-3. Which object layer for user-configured components to provide technical parameters. The ontology layer offers standardized components concept description and relationships with other concepts, and ultimately for the user to retrieve the components required to provide support, but also for the designer body parts distribution services provides a standardized. The business layer provides information as well as manufacturing component designers design components through the management information platform to provide users with the best service parts database.

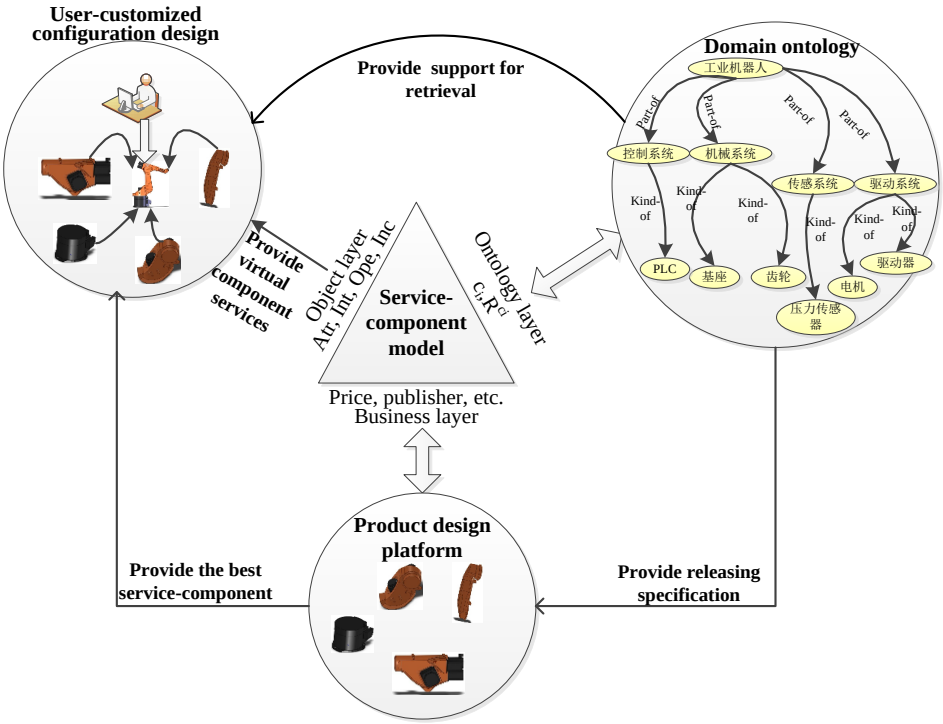


Figure 3-3 The role of service-component

3.3.3 The service-component instantiation

In object-oriented thinking, the process of class assignment to becomes an object is called instantiation. But through formular 2.3 it could be known service-component is not only an abstract class in object-oriented thinking, and also an concept in ontology. So there are two meanings for the instantiation in this article: object instantiation and component instantiation. The definitions are as follows:

(Definition 3.1) Object instantiation: Product designers assign a unique value for each parameter of service-component. This process is called object instantiation. After object

instantiation, the service-component becomes component object, recorded as $_{\Delta}S$.

(Definition 3.2) Component instantiation: Component manufacturers manufacture the component object to he physical component in real world. This process is called component instantiation. After component instantiation, the component is added to instance collection of ontology and stored as knowledge.

The instantiation process of service-component is shown in figure 3-4. Object instantiation enables users to design producr using the component object in the computer, but the component object is still an abstract model present in the component. The component instantiation enables users to obtain the actual physical product. And by adding the instance to ontology instance collection, helps users can quickly find the needed components.

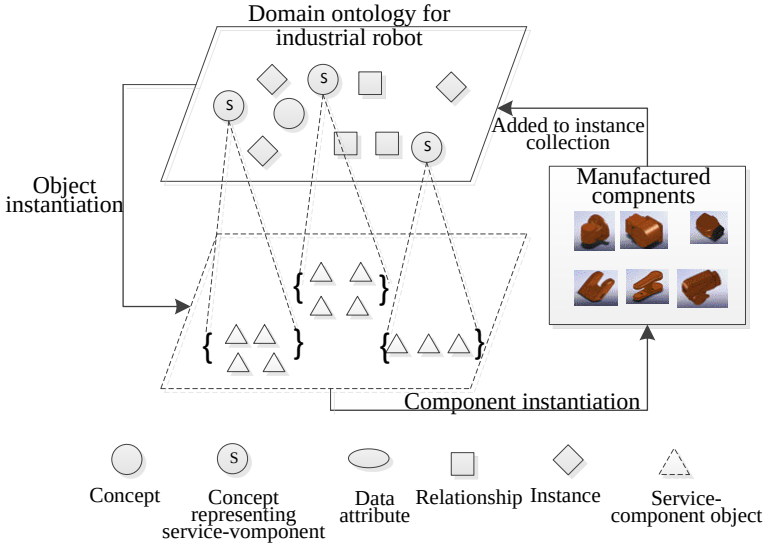


Figure 3-4 The instantiation process of service-component

3.4 The formation and extension of service-component

3.4.1 The formation process of service-component

From the object-oriented perspective, the service-component is an abstraction of component object, and the parameter value set of the component objects constitute the value range of the service-component. From the ontology perspective, the service-component is a collection of component instances. Service-component is created by the platform, and the formation process is shown as figure 3-5. The value range is composed by the component objects released by designers and the component instances released by component manufacturers. The specific process includes the following two aspects:

1) Based on user demand on the market, designers design components, describe the components as component objects and then release them to the platform. The platform

abstracts the component objects with the same C_i as one service-component. Each parameter' value range of the service-component is composed by the parameter value of component object. But these component objects have not been manufactured, so there are no actual physical components. The physical components can be obtained after the component instantiation, as shown by the solid line arrows in figure 3-5;

2) Component manufacturers describe the manufactured components as component objects in accordance with the criteria of the platform, and then release them to the platform. The platform will classify these component objects as a service-component. If the object does not belong to any service-component that already exists on the platform, the platform will add a new service-component. The component objects released by the component manufacturers have component instances inherently, so it can be add to the instance set of ontology directly without component instantiation, as shown by the dashed line arrows in figure 3-5.

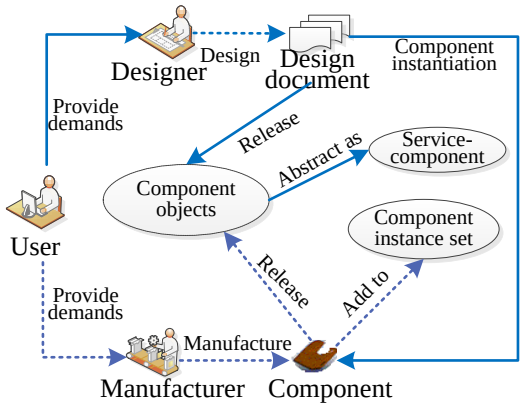


Figure 3-5 The formation process figure of service-component

3.4.2 The extension mechanism of service-component

In the user-customized configuration design process, the user demands for service-component are diverse and constantly changing. Fixed service-component can't provide a good user experience. Service-component needs to be extended appropriately depending on the circumstances. Therefore, the service-component must provide the extension mechanism, but at the same time the extension can't be unlimited. Because unlimited extension may cause unexpected changes in the performance of the service-component,which will result in failure of the configuration design.

The four tuples structure of the service-component is fixed, which ensures the stability of the service-component. The service-component extension starts from object layer. By modifying the technical parameters of the object layer, the service-component realizes the

extension. And the controling of technical parameters extension is realized by the *Inc* inside the service-component. *Inc* sets rules about the extension boundaries of the service-component. The extension of the service-component should follow the rules. The extension has strong adaptability and flexibility while maintaining a stable structure. With the development of new technologies, the extension rules may change, so *Inc* also provides a variable view for external releaser. The service-component can be updated through upgrading *Inc* by the service-component releaser, and there is no need to change other parameters of the service-component.

The extension process is shown in Figure 3-6. The extension is divided into three levels from the bottom up:

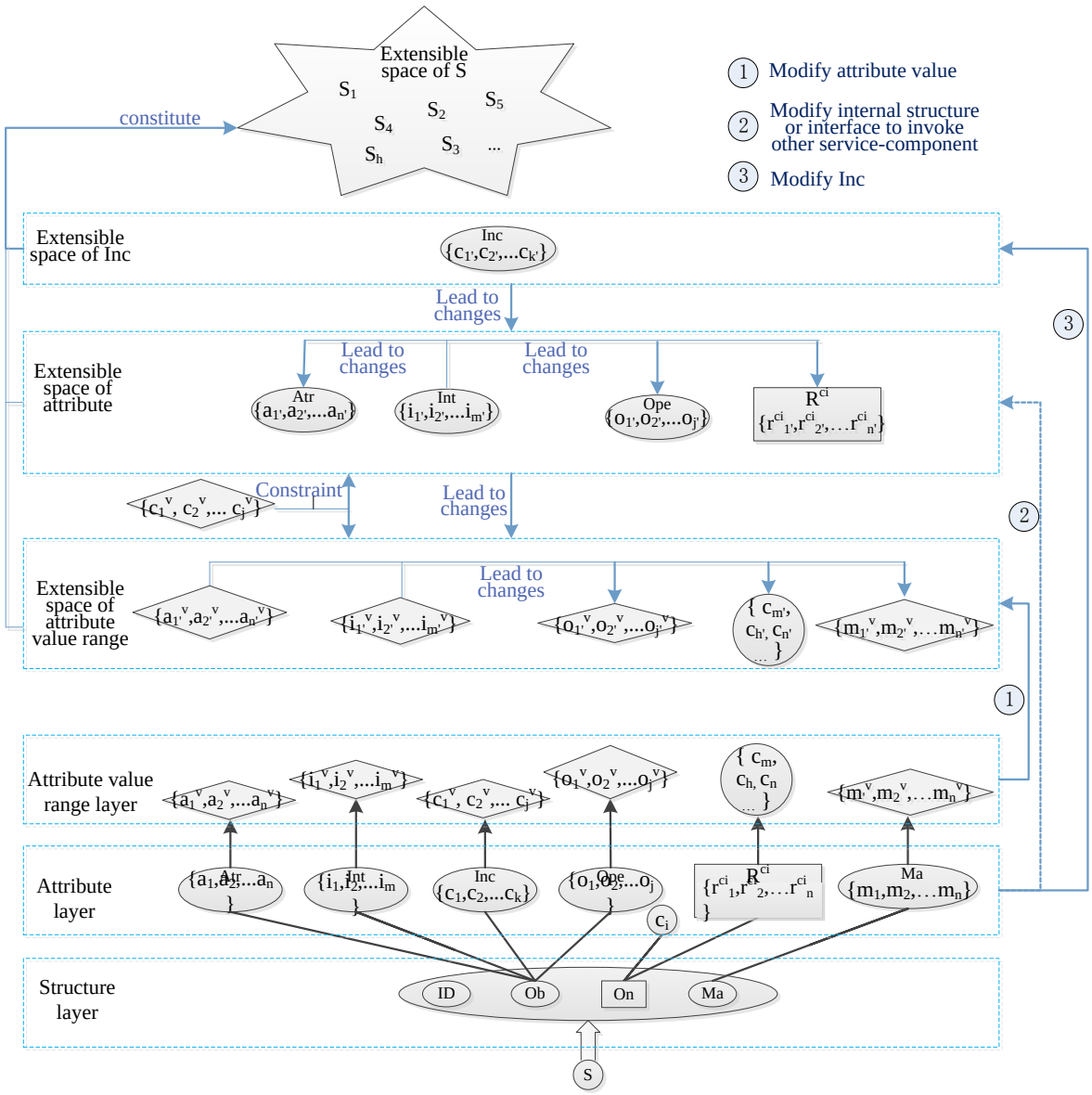


Figure 3-6 Service-component extension

1) It is known from the formation process that each service-component is composed by

component objests. So each attribute of the service-component has a value range. By assigned different values, the service-component can be instantiated to different component objects. When the value range of service-component can't meet the demand, the value range of *Atr* and *Int* in *Ob* need to be extened. The extension essence is to add new component object to the service-component. The value range extension does not vary with the attributes changes of the upper layer in service-component, but it will cause the value range changes of Ope, R^{ci}, Ma in the same layer;

2) When the value range extension can not meet the demand, the attributes of *Int* and *Atr* in *Ob* need to be extended. There are two ways for the attributes extension: one is to realize the interface extension by Int invoking other service-components. For example, the controller can connect the sensor through an interface. By invoking the sensor the controller adds the awareness capabilities and realizes function estension. This extension way is simple and direct way, and it is a prior attribute extension way; the other is to achieve the external interface extension by modifying the internal structure of the service-component. By this way the changes of service-component is large. Only when the interface extension can't satisfy demands will this way be used. Both ways will lead to attribute changes of *Atr*, *Ope*, R^{ci}. When attribute extension is completed, the value range of the attribute in lower has also changed. It should be noted that *Ma* attributes describe some fixed business information, and therefore *Ma* constituent elements can not be extended, but its value range will be changed depending on the value range changing of other attributes. The extension of attributes and attribute value range is not random, and they are constrainted by *Inc*;

3) Finally is the *Inc* extension. When more advanced design method or manufacturing technology appears, the constraints that service-component must comply with may be reduced, and then lead to the changes of *Inc*. The *Inc* extension will not cause a change in service-component, but it brings changes of extensible space for service-component attribute. The attribute extension will lead to a greater changes on value range of attributes. The amplification effect layer by layer makes the *Inc* extension to have the largest influence on service-component. Therefore, the *Inc* extension is rarely used. Usually the extension of the first two layers can meet demands.

User can extend the attribute and the attribute value range. The *Inc* extension is only be taken by the service-component releaser. The three levels is extended from the shallower to the deeper, and make up the service-component extensible space together.

3.5 Case Study: Industrial robot ontology and core components modeling

3.5.1 Industrial robot ontology modeling

Steps to build the ontology include determining the concepts, the level of concepts, the relationship of the concepts and adding instances [149]. Concepts in the field of industrial robot can ontained by enumerate through consulting experts. The concepts of the industrial robot domain established in this thesis consist of three categories, one category is industrial robot types divided according to the function, including welding robot, handling robot, palletizing robot, clean robot, coating robot; the second category is the components constituting an industrial robot, including general mechanical system, electrical system, drive system, sensor system, motor, actuator, base, pillar, joints, as well as some unique end finger; the third category is the functions provided by industrial robots and its components, including providing power, increasing torque, welding, handling, etc. As shown in Figure 3-2, each concept of the ontology contains data attributes and relation attributes.

There are four basic relations in ontology, including the whole and the part relation (part-of), the father and son relation (kind-of), the concept and attribute relation (attribute-of), the concept and instance relation (instance -of) [150]. According to the characteristics of different areas, it can have other relations. Combined ontology and object-oriented thinking, this thesis will determine the relation of industrial robot domain as the following eight types:

Ha: represents *has-a* relation. Ha(C_x,C_y) represents that C_y is a non-essential part of C_x. E.g. Ha (industrial robot, infrared sensor), represents that infrared sensor is part of an industrial robot, but not all industrial robot has an infrared sensor. This relation can also be expressed by -Ha (infrared sensor, industrial robot) ;

Ca: represents *contains-a* relation. Ca (C_x, C_y) represents that C_y is the essential part of C_x. E.g. Ca (industrial robot, base), indicates that any industrial robot must contains the base;

As: represents *associate-of* relation. As(C_x,C_y) represents that there is a necessary business logic link between C_y and C_x outside the four relations above. E.g. As (motor, actuator), indicates that the motor and the actuator need to be used together;

At: represents *attribute-of* relation. At(C_x,C_y) represents that C_y is the arrtibute of C_x. E.g. Ar (base, weight) represents the weight is an attribute of the base;

Io: represents *instance-of* relation. Io(C_x,C_y) represents that C_y is an instance of C_x. E.g. Io (motor, XX type motor) represents XX type motor is an instance of the motor;

Ac: represents *achieve-of* relation. Ac(C_x,C_y) represents that C_y is the realization of C_x.

E.g. Ac (change torque, transmission) represents transmission can realize the function of changing torque;

Do: represents *depend-on* relation. Do(C_x,C_y) represents that C_y depends on C_x. This is a dependency relation between instances with a higher degree of coupling than As relation. E.g. Do (XX type motor, XX type actuator) represents XX type motor must be mapped to XX type actuator;

Ia: represents *is-a* relation. Ia(C_x,C_y) represents that C_y is a sub-concept of C_x. E.g. Do (sensor, infrared sensor) represents infrared sensor is a type of sensor.

In this eight relations, *Ha* and *Ca* all belong to the relation of *part-of*. The association strength of whole and part can be desined more precisely by dividing this relation into two types, so that it can provide more accurate conceptual semantics for the user-customized configuration design. The first seven relations belong to the horizontal relations between concepts, and the last relation is a vertical hyponymy relations between concepts. The concept level in domain can be cleared by defining the concept hyponymy relations.

When creating industrial robot ontology, the concept level i.e. hyponymy relation neens to be cleared first. In protégéclear the concepts with the farther and son relation are added, and then the concept level of industrial robot domain ontology is formed as shown in figure 3-7. Then, the horizontal relations between concepts are analyzed, and then the seven horizontal relations are added to the concepts. Finally, instances is added to the concepts, and the industrial robot domain ontology is shown in figure 3-8. The parts with ◆ markrepresents instances. Most concepts in industrial robot domain ontology have more than one instances. Limited to the space, they are no longer listed the figure.

The original industrial robot domain ontology is established by summing up experiences and expert knowledge. In actual use process, some problems may occur due to the different realities. So that the ontology needs to be modified and improved repeatedly, and ultimately domain ontology can describe the real world t accurately.

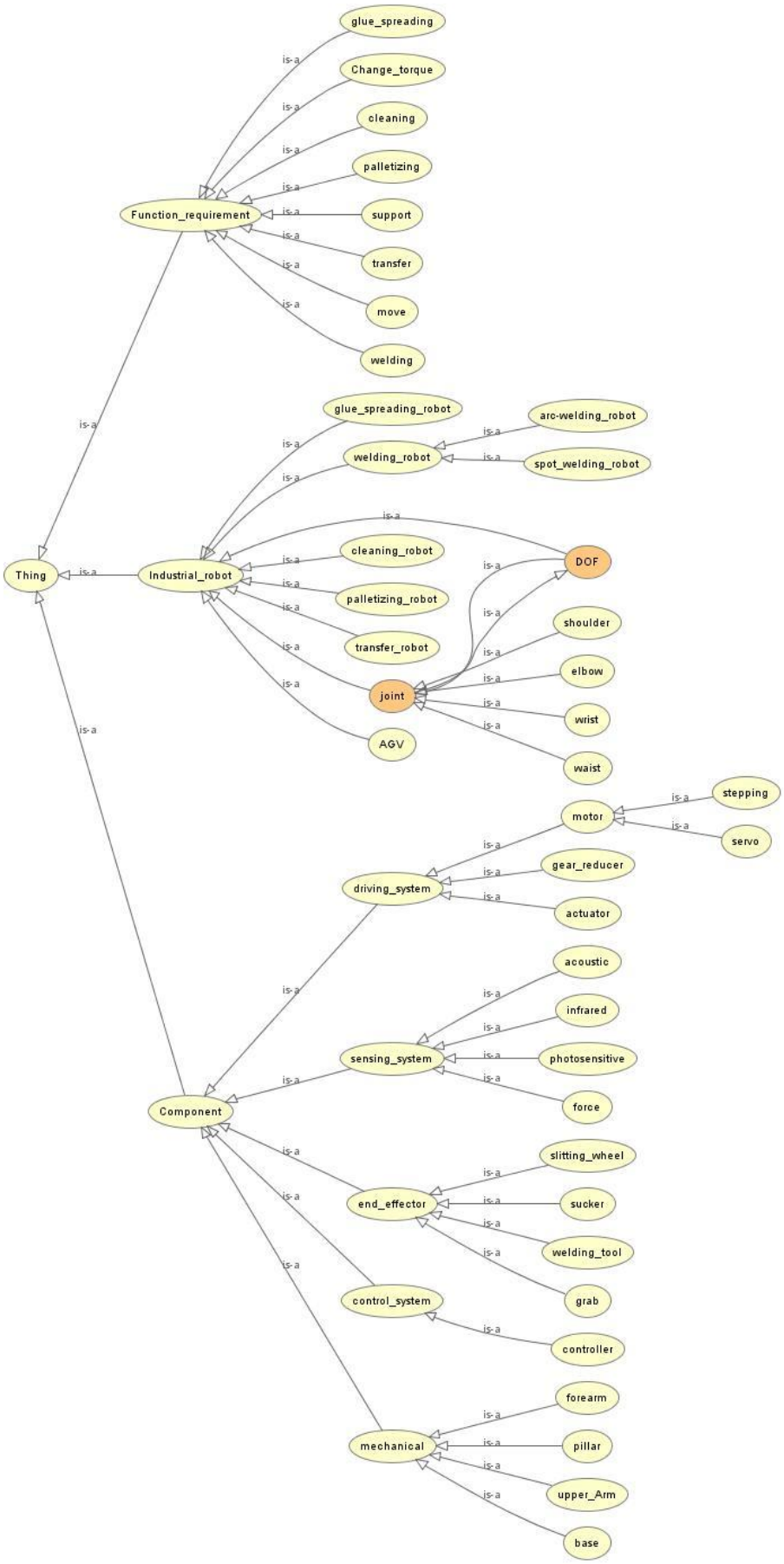


Figure 3-7 The concept level of industrial robot domain ontology

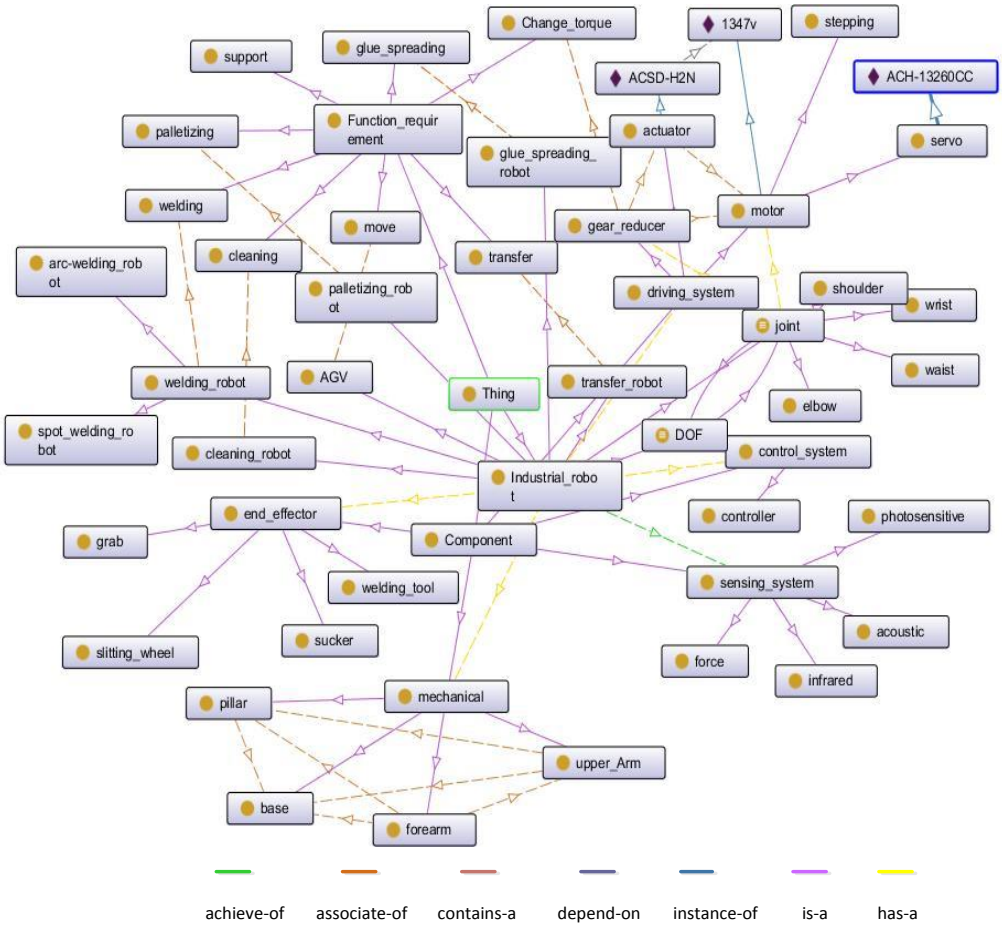


Figure 3-8 Industrial robot domain ontology

3.5.2 Service-component servitization for industrial robot

The following examples show how to formally describe the components of industrial robots as service-component and how to instantiate the service-component in to the specific component objects.

Example 1: harmonic reducer formal description

This concept of reducer exists in ontology, and the concept stores various attributes and attribute value range of the reducer. According to the needs of the market, a designer designed a new harmonic reducer for arm joint of industrial robot. The designer described the harmonic reducer according to the service-component standard as shown in Figure 3-9.

After described, the result can be released to the platform as a component object. Platform will add the parameter value of this component object to the value range of the reducer in ontology. In the future configuration design process, this component object can be invoked by user as object instantiation result. Since the component instantiation process has not been taken for this component object, the manufacturing-related information in *Ma* is empty. If the users select this component object in design process, they can’t get the physical

product.

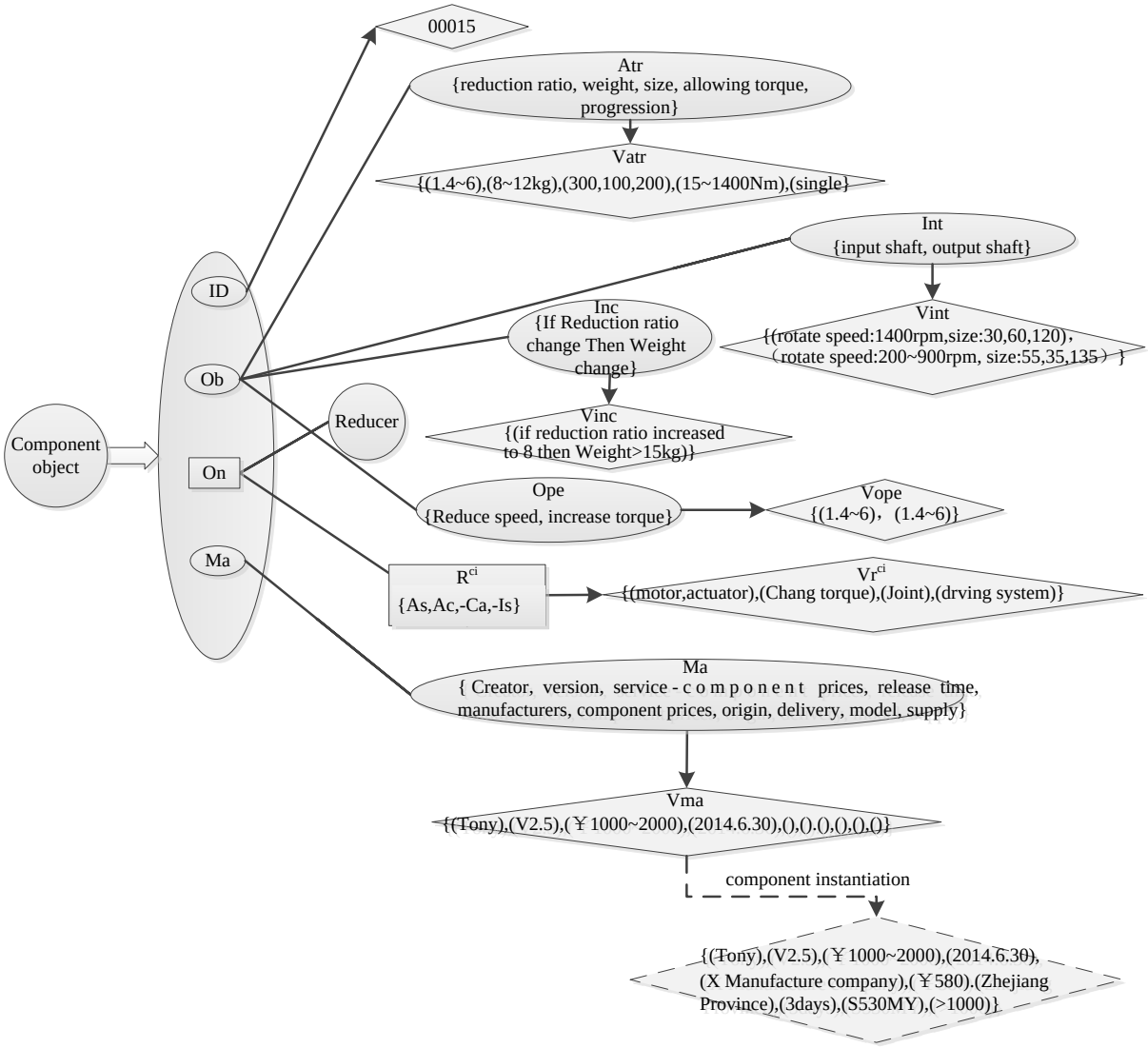


Figure 3-9 A description examples of harmonic reducer

Suppose a manufacturer chose the above reducer in component objects of platform to produce according to their production capacity, that means the component object will become a specific component through the component instantiation process. After produced, the manufacturer, component price, origin, delivery, type, the supply of the factory in *Ma* are assigned to x company, ￥580, Zhejiang Province, 3 days, S530MY, > 1000. The platform stores this component to the reducer instance set of ontology as an instance. If the users select this component instance in design process, they can purchase it directly.

Example 2: Base service-component instantiation extension

User invokes the service-component of base in the design process. The value range of the size parameter in ontology is {height (20,30,40), radius (10,15,20), thickness (5,8,10)}, as

shown in figure 3-10. For simplicity, the figure only indicates the parameters related to extended. By bbject instantiation, three base object size is get: {(20), (10), (5)}, {(30), (15), (8)}, {(40), (20), (10)}, which is shown in figure 3-11.

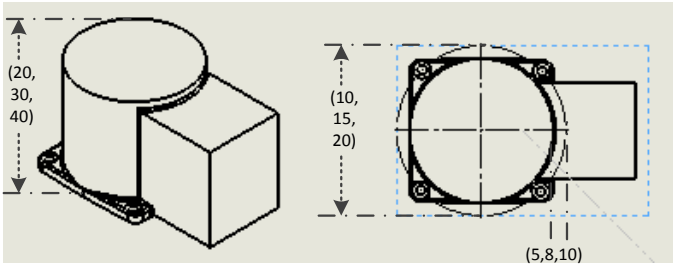


Figure 3-10 The value range of base

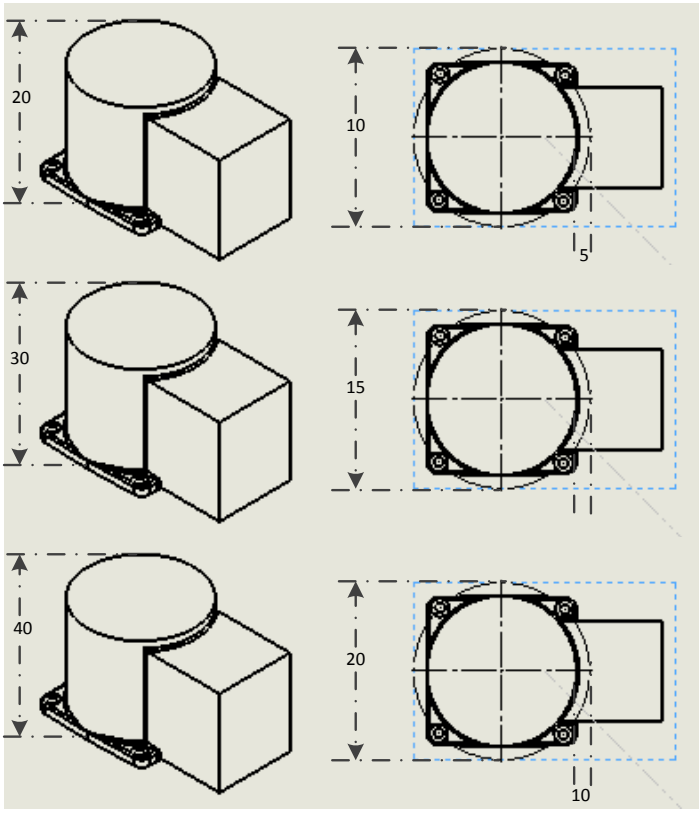


Figure 3-11 The base object after object instantiation

For some users, the height of these three objects can't meet the requirements. Therefore, the users extend base service-object on attribute value range level. The third base object whose size is closest to the requirements is extended, and its height is extended to 45. In order to ensure the support strength of the base, the *Inc* parameter of the base object specifies the relationship of the height and thickness, i.e., whenever the size of height increase 1, the thickness increase 0.1. Therefore, the thickness of the base after extended also changes to 10+ (45-40) * 0.1 = 10.5. A new base object after extended is get, and its size is {(45), (20),

(10.5)}. Subsequently, the users carry out object instantiation on the wxtended service-component by assigning to other parameters of the service-component one by one. Then a extended base is get as shown in Figure 3-12, and users can invoke this component object to design.

After the user completing service-component extension, platform needs to update the service-component of ontology according to the extension result. The value range of size of base is updated to {height (20,30,40,45), radius (10,15,20), thickness (5,8,10,10.5)}. According to the service-component extension mechanism of figure 3-6, it can be known that the extension of *Atr* may lead to the extension of *Ope*, R^{ci}, *Ma*. In this example, the increase of the size range of the base, results in the attribute "corresponding motor type" of *Ope* is extended, and a larger size of the motor type is added to the attribute "corresponding motor type".

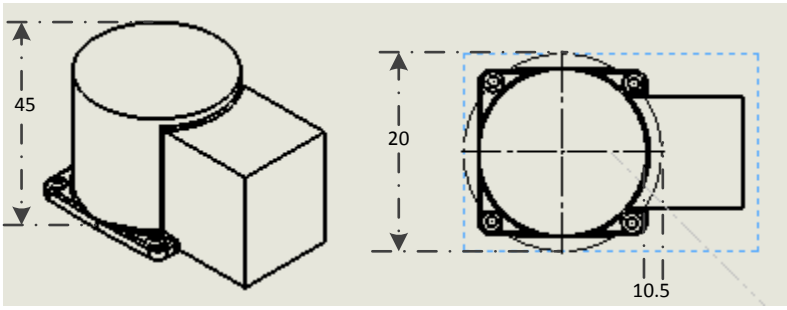


Figure 3-12 The extended base object

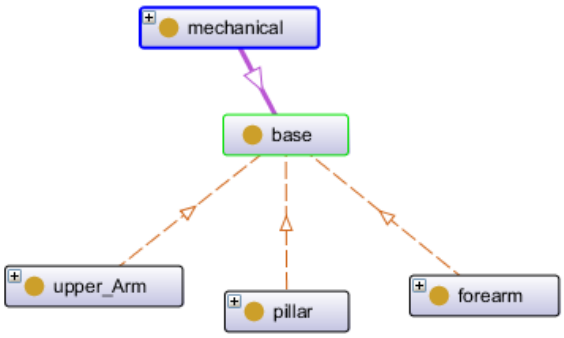


Figure 3-13 The value range of base relation attribute

Similar to the component objects released by designer, there is no instance of the extended base object in the ontology instance set. The physical base only can be get after component instantiation. The extended base manufactured by a manufacturer company is shown in Figure 3-14. This base is added to the ontology instance set. The platform adds the component prices, manufacturer company and other manufacturing-related information of the

instance to the value range of *Ma*. So that it can be invoked by users in the future.

图 3-14 组件实例化后的基座

3.6 Chapter Summary

The component-based theory and method are studied, including the formal description of things, ontology representation, componentization and servitization. On this basis, the componentization description model is established for the product parts. And the model is represented as service-component. The service-component includes ontology layer, object layer, business layer. The object instantiation and component instantiation of the service-component can support the users design process. Next, the formation process and extension method of service-component are introduced. Finally, an example of industrial robot components modeling is analyzed. Firstly industrial robot domain ontology is established by protégé, and then the description method, instantiation process and extension method of service-component are illustrated.

CHAPTER 4 USER-CUSTOMIZED CONFIGURATION DESIGN TEMPLATE FOR INDUSTRIAL ROBOT

To implement user-customized configuration design, only the predefined service-component library is not enough. What stored in service-component library stores is professional technical parameters of service-component, but the user-customized configuration design belongs to system-level design, and it is not possible for the designer to understand the underlying constraints between service-component. Therefore, the user-customized configuration design template for industrial robot is given in this chapter. As a bridge to connect service-components and users, the template can guide users to take configuration design activities. The structure of industrial robot is relatively fixed which offers the possibility for the establishment of user-customized configuration design template.

4.1 Industrial Robot Composition Structure

Industrial robot is composed of mechanical system, drive system, control system and sensing system. As shown in Figure 4-1, the contents of the dark box represent different types of the same layer, and the contents of light-colored box indicate different constituent parts of the upper layer.

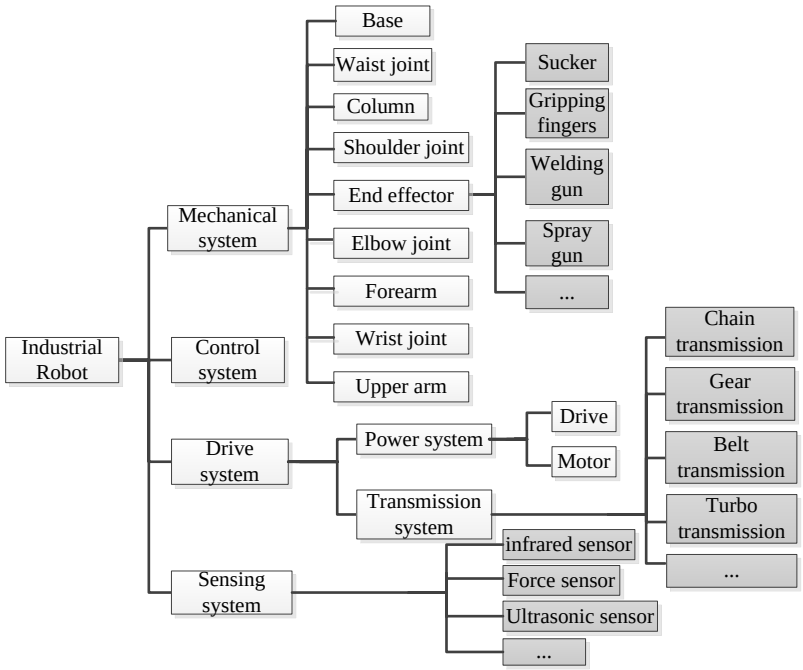


Figure 4-1 Industrial robot composition

From the figure, it shows that the mechanical body includes base, pillar, upper arm, forearm and end effector. Depending on the task, the end effector is different, such as sucker,

gripping fingers, welding gun, spray gun, etc. Drive system includes power system and transmission system. Power systems including drive, motor, and transmission system contains chain transmission, gear transmission, belt transmission, turbo transmission and so on. Control system does not belong in the content of configuration design which is studied in this paper, so it doesn't be researched here. Sensing system includes infrared sensor, force sensor, ultrasonic sensor, etc. The sensor can be selected according to the needs of the robot to make robot has a certain awareness capability.

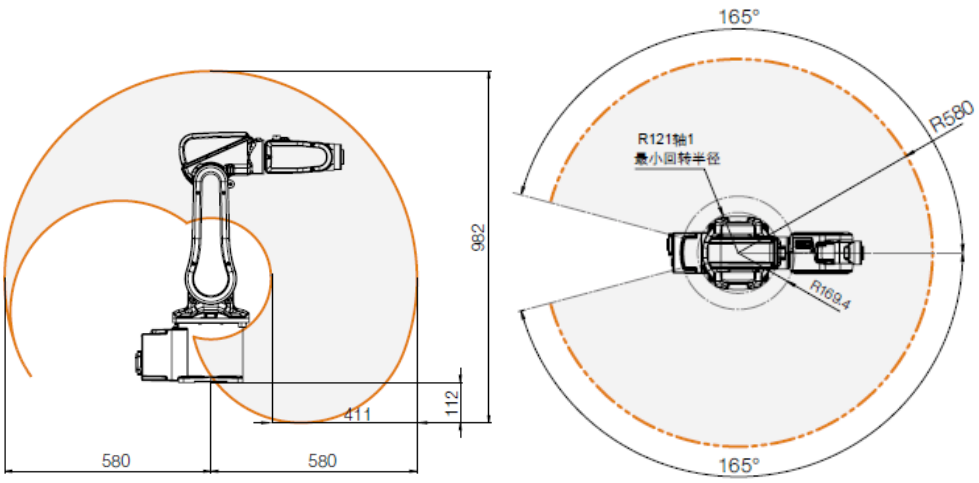
The existing of some common industrial robots are shown in Figure 4-2 (a). Between the base and the pillar is waist joint; between the pillar and the upper arm is shoulder joint; between the upper arm and forearm is elbow joint; between the forearm and the end effector is wrist joint. Point *P* is the wrist reference point. Driven by the drive system, the waist joint rotates around the z-axis, and .the shoulder and elbow joint rotates around the y-axis. Waist joint, shoulder joint, and elbow joint form three positions degree of freedom (DOF) of industrial robot, and wrist joint have 0 or more posture DOF. The more DOF the industrial robot has, the more flexible of the industrial robot movement. But it also will lead to the complexity of control. Therefore for the degree of freedom, more is not better. Usually 3 to 6 degrees of freedom can meet most requirements. By pose transformation, industrial robots can move in space and perform different tasks. In the process of moving, the set of point that *P* can reach constitutes the industrial robots' working space. Work space is closely related to the angular range that industrial robots' joints can reach. With ABB's IRB120 for example, the published work space is shown in Figure 4-2 (b) below.

KUKA spot welding robot KR180-2

KUKA small scale robot KR5

ABB robot IRB120

(a) the existing of some common industrial robots



(b) industrial robots' working space

Figure 4-2 The existing of some common industrial robots and working space

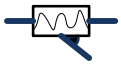
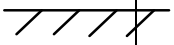
4.2 User-customized configuration design template establishment

The configuration design template is established according to the structure of industrial robots. And it provides designers straightforward view in the form of a mechanical schematic.

4.2.1 Template graphic elements

From a mechanical point of view, industrial robot is an open chain structure composed by rod-shaped parts which are connected by kinematic pair. In the user-customized configuration design process, users only configure the whole structure and the movement pattern of kinematic pair, and not design the specific structural of components. Therefore, the mechanical schematic which ignores the specific structure of component is taken to establish the template. The mechanical schematic symbols used by the template are shown in table 4-1. In table 4-1(a), the symbols are taken from ISO3952/1-1981, ISO3952/2-1981, ISO3952/3-1979. And the symbols in (b) are needed here but not taken from ISO.

Table 4-1 Template symbols from ISO (a)

Components of industrial robot	**ISO3952 standard symbol**	**Components of industrial robot**	**ISO3952 standard symbol**	**Components of industrial robot**	**ISO3952 standard symbol**
Rotation pair		Plane rotation pair		Translation pair	
Screw pair		Shaft		Rack(fixed rack)	

Continued from Table 4-1

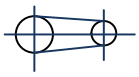
Components of industrial robot	**ISO3952 standard symbol**	**Components of industrial robot**	**ISO3952 standard symbol**	**Components of industrial robot**	**ISO3952 standard symbol**
Coupler		Belt/chain transmission		Gear transmission	
prime motor (universal)		Electric motor		Bearing	

Table 4-2 Template symbols (b)

Components of industrial robot	Symbol	**Components of industrial robot**	Symbol	**Components of industrial robot**	Symbol
End effector		Actuator		Sensor	

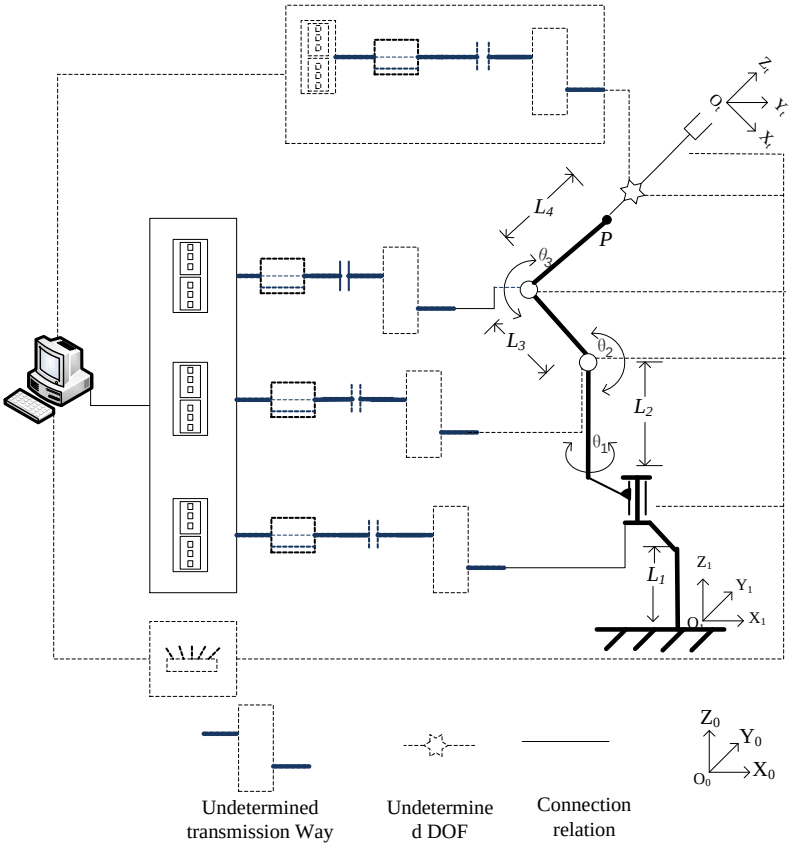


Figure 4-3 User-customized configuration design template

The industrial robot user-customized configuration design template is shown in figure 4-3. The template contains basic elements of building an industrial robot. Each joint of the

robot contains a prime motor, a reducer and an actuator corresponding to the prime motor. Actuator is usually installed outside of the robot, and is connected to the motor by control line. The contents in the dotted box represent undetermined elements, and they will be determined by users, then the dotted line will become solid line. For example, the undetermined transmission way can be belt transmission, gear transmission, chain transmission and so on. The undetermined DOF is wrist joint DOF. The sensor can be in installed in some part of the robot as needed. L1 ~ L4 represent the lengths of base, pillar, upper arm, forearm respectively. O_0-X_0-Y_0-Z_0 is the absolute coordinate system. It takes the earth as a fixed reference, and it is regardless of the position and orientation of the robot. O_1-X_1-Y_1-Z_1 is the base coordinate system. It takes the base surface o of the robot as reference, and the default is generally the same as the absolute coordinate system. O_t-X_t-Y_t-Z_t is the tool coordinate system. It takes the end effector as reference, and it provides a reference for the end effector relative to the robot body. θ_1~θ_3 represent the angles of the three position DOF. The base coordinate system is adopted to the three angles, and the positive rotation is counter-clockwise.

4.2.2 Template Parameters

According to the content of the dotted box in figure 4-3, the parameters of template needed to be determined are as follows:

$$P_s=\{N_{DOF}, T_{KP}, T_{DS}, T_{Tr}, T_{Se}, T_{EE}, W_{EE}, L_a\} \tag{4.1}$$

N_{DOF} is the number of DOF;

$T_{KP}=\{ T_{KP1}, T_{KP2},... T_{KPn}\}$, T_{KPi} is the kinematic pair type of the i*th* DOF, *n* is the number of DOF;

$T_{DS}=\{ T_{DS1}, T_{DS2},... T_{DSn}\}$, T_{DSi} represents the prime motor type which provides motive for T_{KPi}. It is usually a motor except under exceptional circumstances;

$T_{Tr}=\{ T_{Tr1}, T_{Tr2},... T_{Trn}\}$, T_{Tri} represents the transmission way used by T_{KPi}. It is usually a gear reducer except under exceptional circumstances;

$T_{Se}=\{ T_{Se1}, T_{Se2},... T_{Sem}\}$, T_{Sei} 为 represents the type of the i*th* sensor, *m* is the number of sensor used in the template;

T_{EE} is the end effector type, and it is determined by the robot task;

W_{EE} is the maximum load imposed on the end effector;

L_a is the minimum length of the robot arm, and it determines the work space of the robot。

P_s represents the static parameters needed to be determined in figure 4-3. In addition to

these, some dynamic parameters need to be input to the template. The dynamic parameters P_d are represented as follows:

$$P_d=\{\ \omega_p, A_p, \omega_{ee}, A_{ee}\ \} \tag{4.2}$$

ω_p represents the velocity of point P;

A_p represents the accelerated velocity of point P;

ω_{ee} represents the velocity of end effector;

A_{ee} represents the accelerated velocity of end effector.

After determining the template parameters, it will become an industrial robot template composed by service-components. Through value assignment to the service-components, the template will become a specific product composed by service-component objects.

4.2.3 Internal constraint of template

In the value assignment process of service-components, the following constraints should be followed:

$$T_i * \omega_i < T_{KPi}.P \quad \cdots\cdots\cdots \tag{4.3}$$

$$T_{KPi}.N:\ N_{load}=i \quad \cdots\cdots\cdots \tag{4.4}$$

$$T_{Tri}.\ i_{min} \leqslant i \leqslant T_{Tri}.i_{max} \quad \cdots \tag{4.5}$$

$$T_{KPi}.T_{rated} * i \leqslant T_{Tri}.\ T_{rated} \cdots\cdots\cdots\cdots \tag{4.6}$$

$$5 \leqslant (J_{load}/i^2)/\ T_{KPi}.J \leqslant 20 \text{（Non-fixed value, and can be adjusted according to user's needs）} \quad \cdots\cdots\cdots \quad \cdots \tag{4.7}$$

$$L_a \leqslant L_3 + L_4 \cdots \quad \cdots\cdots\cdots \tag{4.8}$$

If S_i *has connection relation with* S_j

$$\text{Then } \exists S_i.Int_x = S_j.Int_y \quad \cdots \tag{4.9}$$

Where the respective symbols have the following meanings:

T_i: The maximum torque of the i*th* DOF, and the value is given by the technical model in next section;

ω_i:The angular velocity of the i*th* DOF;

$T_{KPi}.P$: The prime motor's power of the i*th* DOF;

$T_{KPi}.N$: The prime motor's rotate speed of the i*th* DOF;

N_{load}: Load speed (n/min);

I:reduction ratio;

$T_{Tri}.\ i_{min}$: The transmission's minimum reduction ratio of the ith DOF;

$T_{Tri}.i_{max}$: The transmission's maximum reduction ratio of the ith DOF;

$T_{KPi}.T_{rated}$: The rime motor's rated torque of the ith DOF;

$T_{Tri}.\ T_{rated}$: The transmission's rated torque of the ith DOF;

J_{load}: The load moment of inertia (kg·m²);

$T_{KPi}.J$: The prime motor moment of inertia of the ith DOF;

The formula (4.9) means that there is at least one matched physical interface between two connected sercive-components.

Internal constraints of template are the basic constrained which must be met between the components. They guarantees the feasibility of the final configuration result, and other constraints set by users to industrial robot must be able to meet these basic internal constraints first.

4.3 Kinematics and dynamics analysis of template parameters

Configuring product through template ensures that the static parameters of industrial robot meet the user's requirements. But it can't guarantee that the dynamic characteristics of the final product can meet the requirements, such as the required torque of joint rotation should be calculated according to the user's velocity and acceleration requirements. Therefore, the corresponding technical model for template is needed to analyze the dynamic parameters of industrial robot, including work space solving, kinematics and dynamics analysis, etc. As the dynamic characteristics constraints of service-components, the analysis result could provide reference for user-customized design process. The kinematics and dynamics analysis software commonly used contains Simscape, Adams, Pro / MECHANICA and so on. In this paper, Simscape is used for the kinematics and dynamics analysis.

The Simscape is a platform for multi-domain modeling and simulation of physical systems launched by MathWorks company. It can build model for a variety of physical fields of mechanical, electrical and others using physical blocks, and obtain the desired results fastly without building complex calculus equations manually. When the corresponding technical model for user-customized configuration design template is being built, the required Simscape modeling elements are as shown in Table 4-2.

Table 4-2 Simscape model elements

Figure	Meaning
Body	Represents the rigid bodies of base, upper arm, etc in the template.
Disassembled Revolute	Represents the revolute pair. The two rigid bodies connected by this can make relative revolution around a particular axis.
Joint Actuator	The element which can applied force, angular velocity, torque to the joint.
Joint Sensor	The element which can measure torque, angular velocity of the joint.
Body Actuator	The element which can applied force, angular velocity, torque to the rigid bodies.
Body Sensor	Element for measuring rigid, rigid measure the force, speed, etc. The element which can measure force, velocity of the rigid bodies.
Ground	A stationary point in absolute coordinates. There must be one rigid body connected to Ground in the built model, and it is means the model is fixed in an inertial system.

The technical model of the template in figure 4-3 is established using Simscape as shown in Figure 4-4. θ1~θ3 represent waist joint, shoulder joint, and elbow joint. When the user-customized configuration design template does not enter any parameters to the model, the input value of each *Joint Actuator* is empty, and the output value of each *Joint Sensor* is also empty. The technical model only contains the first three degrees of freedom. Because before configuration is completed, the wrist joint and the end effector can't be determined.

The purpose of the modeling is to determine workspace, simulate kinematics and dynamics behavior by parameters setting. The simulation result can provide a reference for the designer to configure. The needed results of model simulation can quickly and intuitively be obtained by Simscape , without manual definition of complex calculus equations.

Before dynamics simulation, the moment of each rigid body's inertia is needed to calculate. But before all the service-component parameters are determined, the used components can't be determined, so the moment of the inertia can't be accurately calculated. During the actual design process of industrial robot, the moment of each rigid body's inertia are determined by estimation generally. The estimation in figure 4-4 is based on the following assumptions:

a) All rigid bodies in the model are cylindrical rigid body with uniformly-distributed mass.

b) The prime motor and transmission is fixed in the rigid body's center and coaxial with the rigid body.

c) The mass of prime motor and transmission is uniformly-distributed.

d) After being installed, the prime motor and transmission are regarded as a whole.

e) The energy conversion efficiency in robot running process is 100%.

Based on the above assumptions, the length of the rigid body L is set to l (m), the weight of L is W_L(kg), the weight of the prime motor and transmission inside L are W_D(kg)、W_T(kg) respectively. The formulas calculating the moment of rigid body inertia are as follows:

$$W= W_L+W_D+W_T \tag{4.10}$$

The moment of rigid body inertia around the y-axis J_y(kg·m²):

$$J_y=1/3Wl_4^2 \tag{4.11}$$

The moment of rigid body inertia around the y-axis J_z(kg·m²):

$$J_z=1/2Wl^2 \cdots \tag{4.12}$$

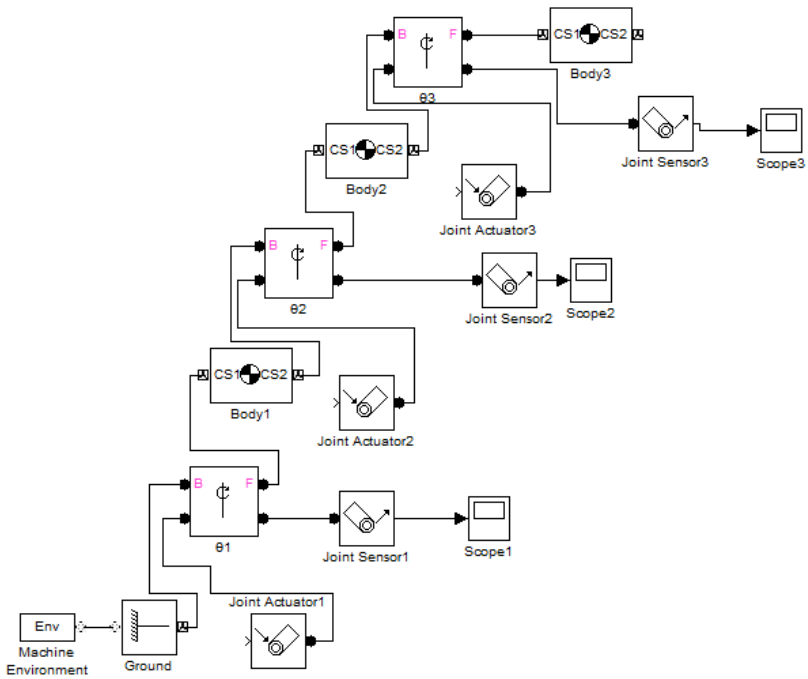


Figure 4-4 Technical model

The value of W_D、W_T in Formula (4.10) can be obtained by querying the weight estimate table. Based on the experts' experiences, the weight estimate table records the weight range of the drive system(including prime motor and transmission) and rigid bodies corresponding to the load, as shown in Table 4-3. The technical model estimates the moment of inertia of rigid bodies based on the values in the weight estimate table, and therefore the accuracy of the table will determine the exact extent of the technical template parameter estimation. The table was originally given by domain experts based on their experience, and constantly revised in the

process of using, thereby increasing the accuracy of the values in table.

Table 4-3 Weight estimate table

load range (kg)	**weight range of drive system in position DOF (kg)**	**weight range of drive system in posture DOF (kg)**	**weight range of rigid body(kg)**
0~10	20~35	15~17	5~15
10~20	30~46	20~35	10~35
…	…	…	…

After determining the moment of inertia, technical template can calculate the required torque for each joint, and the calculation formula is as follows. *T* is a joint torque when rotating (N · m), *J* is the moment of inertia of the rigid body driven by the joint axis (kg·m²), and ω is the angular accelerated velocity of joint movement (rad/s^2):

$$T=J*\omega\cdot\cdot \tag{4.13}$$

4.4 Finite Element Analysis of template Parameters

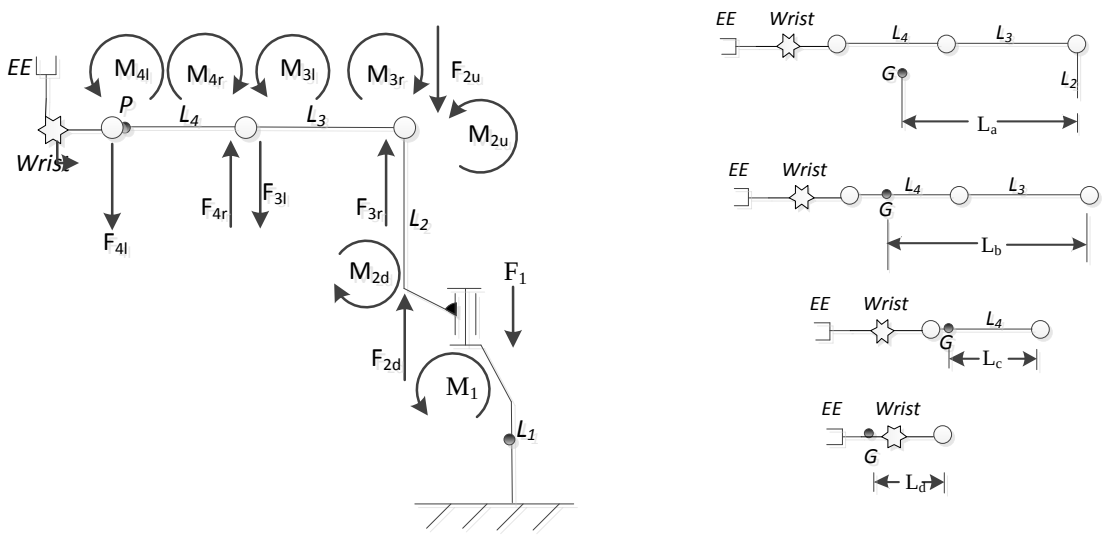
Finite element analysis (FEA) refers to decompose the physical structure into a finite number of elements interacting on each other, set an approximate mathematical solution for each element, and based on this, obtain the solution of the whole mechanical structure. FEA can achieve the simulation of real mechanical structure under different loads, working conditions, constraints and other conditions. The aim of the kinematics and dynamics analysis in section 4.3 is to provide dynamic parameter constraints for the user-customized configuration design, and it begins before configuration design. And in this section, the aim of FEA is to analyze the performance of the user-customized configuration design result, thus providing a basis for improving configuration design result. The FEA begins analysis on configuration template after the configuration is complete.

The industrial robot finite element analysis includes statics analysis and modal analysis. Statics analysis researches the balance condition (including stationary and uniform linear motion) of the mechanical structure under stress. The condition generally refers to the mechanical structure stress and strain in the equilibrium state. Through applying load to mechanical structures, the mechanical structure stress and strain analysis is taken. When the stress and strain of mechanical structure is too large, an appropriate interpolation value can be provided in the control system, or improving the mechanical structure to improve structural strength, thereby improving the motion accuracy of industrial robot. Modal is a basic concept of structural dynamics. It refers to the natural vibration characteristics of the mechanical structure (including frequencies and vibration mode). Mechanical structure has different modal, and each modal has a corresponding frequency and vibration mode. The process of

analyzing and calculating the modal of mechanical structure is modal analysis. The natural frequencies and vibration mode belonging to each modal can be determined through modal analysis, so the actual vibration response of mechanical structure in the role of a vibration source can be known. The finite element analysis software commonly used includes ANSYS, Pro / MECHANICA, ABAQUS, etc. In this paper, the Pro/MECHANICA is chosen.

4.4.1 Statics Analysis

In this paper, the maximum force pose of industrial robots is selected as the statics analysis working condition. When the arm of industrial robot is outwardly stretch in a straight line as shown in figure 4-5, the stress of each arm is maximum, and the load calculation formulas of main structure are shown as 4.14 ~ 4.16 [151].



(a) the maximum force pose (b) centre-of-gravity position of each component

Figure 4-5 Free-body diagram of industrial robot

Load calculation formula of base L1:

$$\begin{cases} F_1 = G_{EE} + G_{wrist} + G_{L4} + G_{L3} + G_{L2} \\ M_1 = F_1 * L_a \end{cases} \quad \ldots\ldots \quad (4.14)$$

Load calculation formula of upper arm L3:

$$\begin{cases} F_{3l} = G_{EE} + G_{wrist} + G_{L4} \\ F_{3r} = F_{3l} + G_{L3} \\ M_{3l} = F_{3l} * L_c \\ M_{3r} = M_{3l} + F_{3l} * L_3 + G_{L3} * L_{G3} \end{cases} \quad \ldots\ldots. \quad (4.15)$$

Load calculation formula of forearm L4:

$$\begin{cases} F_{4l} = G_{EE} + G_{wrist} \\ F_{4r} = F_{4l} + G_{L4} \\ M_{4l} = F_{4l} * L_d \\ M_{4r} = M_{4l} + F_{4l} * L_4 + G_{L4} * L_{G4} \end{cases} \tag{4.16}$$

In the industrial robot template herein, in addition to the three basic structures, further comprises a pillar L2. Its load calculation formula is as shown in equation 4.17:

$$\begin{cases} \mathrm{F}_{2u} = \mathrm{G}_{EE} + \mathrm{G}_{wrist} + \mathrm{G}_{L4} + \mathrm{G}_{L3} \\ \mathrm{F}_{2d} = F_{2u} + G_{L2} \\ M_{2u} = F_{2u} * L_b \\ M_{2d} = M_{2u} \end{cases} \tag{4.17}$$

In formula 4.12 ~ 4.15, the *F* is the force of gravity applied by the connected rigid body, the *M* is torque applied by the *F* to the rigid body, and the *G* is gravity of the rigid body. Note that the weight of each rigid body contains the weight of drive system inside the rigid body, which is calculated by formula 4.8.

After the user-customized configuration design is completed, the service-components in template are assigned values, and then become service-component objects. Each service-component object contains three-dimensional model of the object. The three-dimensional models of the base, pillar, upper arm, forearm, are imported into ANSYS respectively for statics analysis. First, the parameters of the model are set, including the definition of the type of material, applying the load calculated by formula (4.14) - (4.17), and then the stress and strain nephogram for each rigid body is obtained by running the model. Through searching the maximum deformation data, it can be determined whether the material and structural strength of the rigid body meets the accuracy requirements or not.

4.4.2 Modal analysis

The main purpose of industrial robots modal analysis is to check whether there is resonance problem during robot movement, and then take noise reduction processing on the robot. There are many reasons causing resonance, including drive system rotation, shaft rotation, excessive transmission clearance, etc. Generally, the excitation frequency of drive system and the natural frequency of the rigid body are stored in the Ob attributes of service-component as the technical parameters. When the external excitation frequency is greater than the first order natural frequency of the rigid body, it will resonate.

After the user-customized configuration design is completed, each service-components in the configuration template will become components, and the natural frequency of each

component could be obtained by ANSYS simulation. The specific process is similar to Section 4.4.1. Firstly the three-dimensional models of the industrial robot components including base, pillar, upper arm and forearm are imported into ANSYS, secondly the model parameters are set including the element material of solid45, material elastic modulus of 2.2E11, Poisson's ratio of 0.3, the density of 7800, a free partition meshing, the type of analysis to Modal, a frequency range of 1000 to 10000. After all parameters have been set, the *solve* module in *Solution* is used to solve. Each structure has an infinite-order modal, and the low-order modes determine the vibration response characteristics of the structure. So it usually computes the first few order modal only. In this article, only the first five order modal is solved, and the first five order modal and vibration nephogram of the main structure is obtained by ANSYS.

4.5 Template operating principle

Through a series of processing to the parameters inputted, the user-customized configuration design template outputs the components list of industrial robot and performance parameters meeting the needs. The flow and analysis process of template parameters is shown in Figure 4-5.

According to the needs of the task, user inputs the needed parameters to template, including the DOF of wrist joint, the kinematic pair type of each DOF, the workload weight, the end effector type, the minimum arm length, velocity and other parameters. Simscape model obtains parameters such as load weight, arm length, joint angle from the configuration template, simultaneously query the weight estimate table, calculate the parameters of the work space, joint torque and then output them to the configuration template. In the constraint of the Simscape calculation results and configuration template internal constraint, the service-component objects meeting the constraints can be retrieved from the database to compose the configuration result. The detailed retrieval process is given by the chapter 5. When all the service-components of configuration template have become into service-component objects, the corresponding three-dimensional models of objects and the calculated load data will be input to ANSYS. The ANSYS is used in statics analysis and modal analysis of the configuration result. The analysis results including stress-strain data and vibration frequency data. These data is output to template. Finally, the template will send the configuration result and ANSYS analysis result to user.

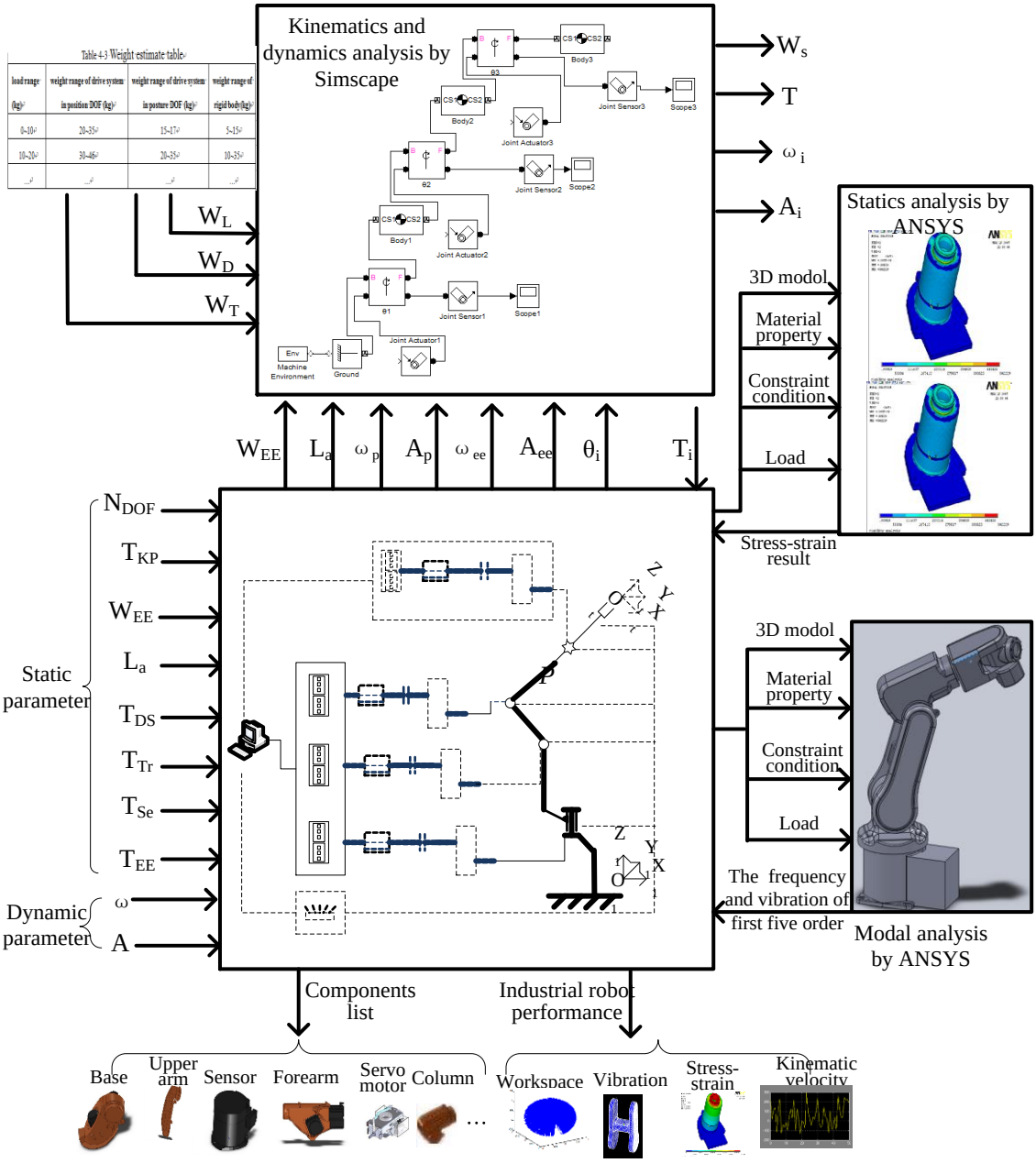


Figure 4-5 Template operating principle

4.6 Template analysis example

(1) Parameter setting for user-customized configuration design template

First, the parameter values are input to the template. The values of P_s, P_d are as follows:

N_{DOF}=5; T_{KP}={spatial rotation, plane rotation, plane rotation, plane rotation, spatial rotation}; T_{DS}={servo motor, servo motor, servo motor, servo motor, servo motor}; T_{Tr}={gear reducer, gear reducer, gear reducer, gear reducer, gear reducer}; T_{Se}={Pressure sensor}; T_{EE}= Negative pressure type sucker; W_{EE}=6kg; The minimum length meeting the requirement in the database is selected for L_i, in this case, L_1~L_4={0.2m,0.3m,0.4m,0.4m}; L_a=0.6m; ω_p=3rad/s,

A_p=6rad/s^2. After the parameter values are determined, the user-customized configuration design template is shown in Figure 4-6.

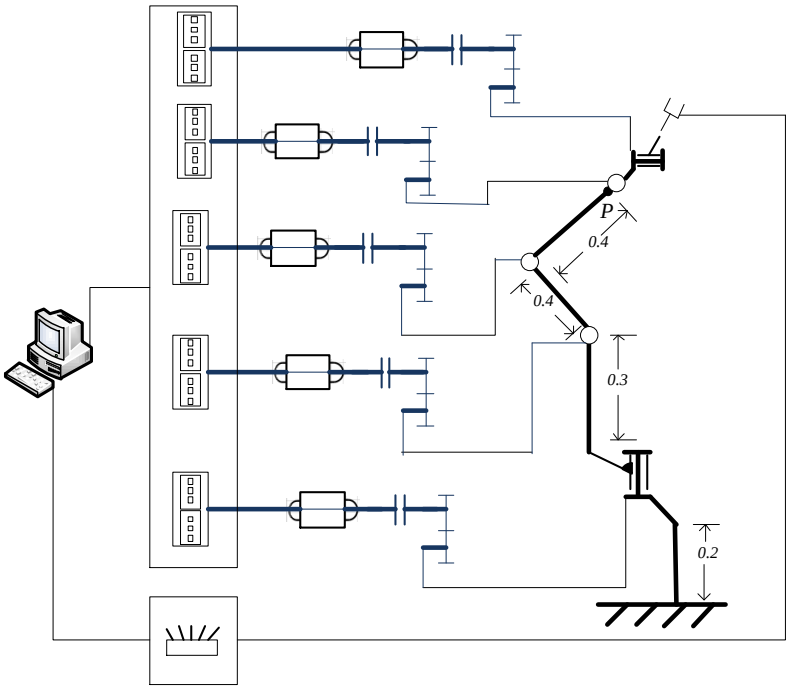
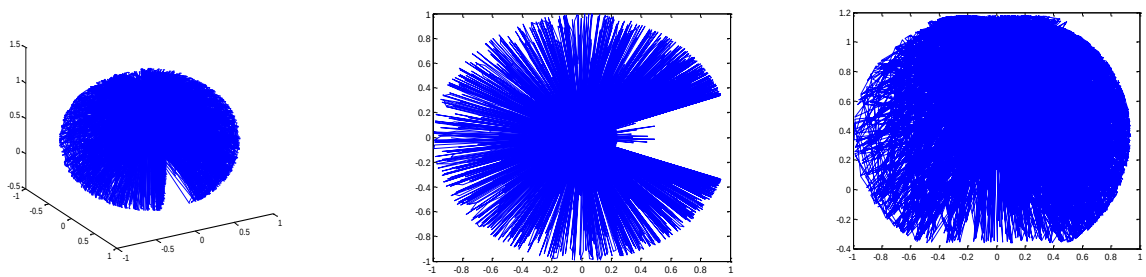


Figure 4-6 User-customized configuration design template after parameters inputing

(2) Kinematics and dynamics simulation

Then the parameters' values in the configuration template are assigned to the Simscape model. Wherein the value of *Body Actuator* is calculated by W_{EE}*a=6*10=60N, the angular velocity of three *Joint Actuator* is set to ω=3rad/s and angular acceleration is set to A=6rad/s^2. Through querying the weight estimation table and formula (4.10) ~ (4.12), the moment of inertia in waist joint, shoulder joint, elbow joint are calculated to 0.6 kg·m^2、3.5 kg·m^2、2.4 kg·m^{2}respectively. The calculated results are input to Simscape model as input parameters, and then the workspace (*Ws*) of the configuration template in different views is shown in figure 4-7. The torque (T_i) in waist joint, shoulder joint, elbow joint is shown in figure 4-8. The parameters of L_i, T_i, ω_i, A_i are input to formula (4.3) ~ (4.9), and the internal constraints of the configuration template are obtained.

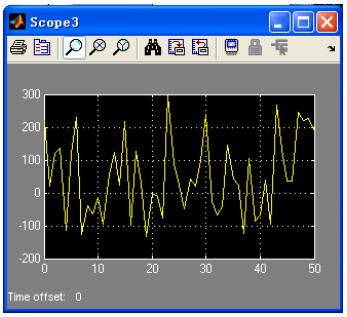


4-7(a) 3D view

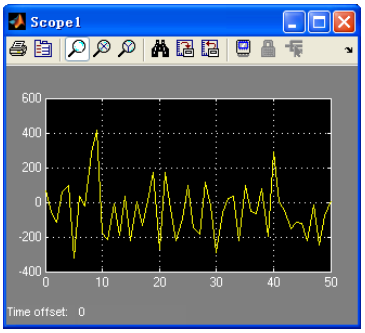
4-7(b) xoy plane view

4-7(c) xoz plane view

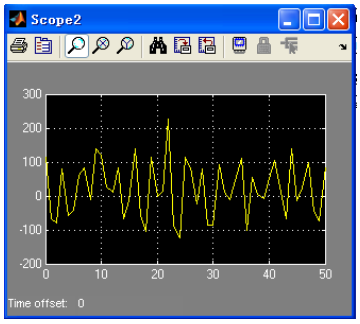
Figure 4-7 Workspace figure



4-8 (a) waist joint



4-8 (b) shoulder joint

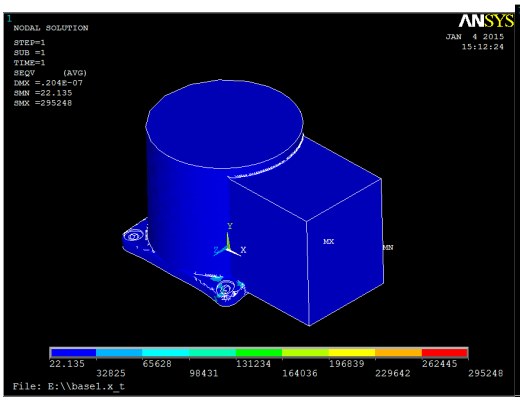


4-8 (c) elbow joint

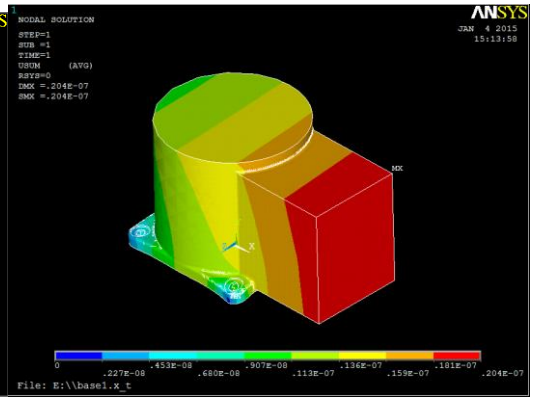
Figure 4-8 The torque of three joints

(3) statics analysis

According to the formula 4.12 ~ 4.15, the stress analysis results of base, pillar, upper arm, forearm are respectively F_1=600N, M_1=360N, F_{2u}=500N, F_{2d}=700N, M_{2u}=350N, M_{2d}=350N, F_{3l}=300N, F_{3r}=500N, M_{3l}=120N, M_{3r}=280N, F_{4l}=100N, F_{4r}=300N, M_{4l}=20N, M_{4r}=100N. Input the calculation results of load, torque and gravity data to ANSYS, at the same time set the element type to solid45, material elastic modulus to 2.2E11, poisson's ratio to 0.3, the density to 7800, and then analyze it. The four main body structure stress nephogram and displacement nephogram are onbtained respectively as shown in figure 4-9~4-12.

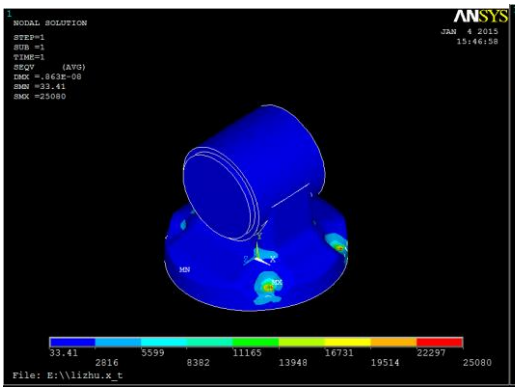


(a) stress nephogram

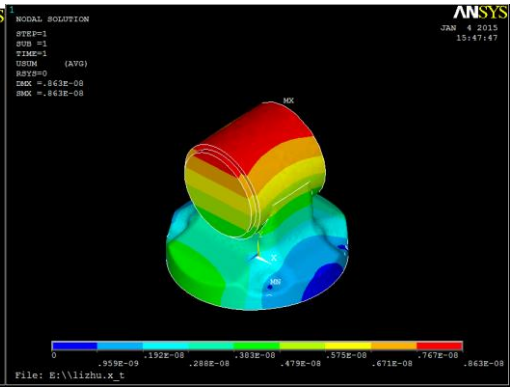


(b) displacement nephogram

Figure 4-9 Statics analysis figure of base

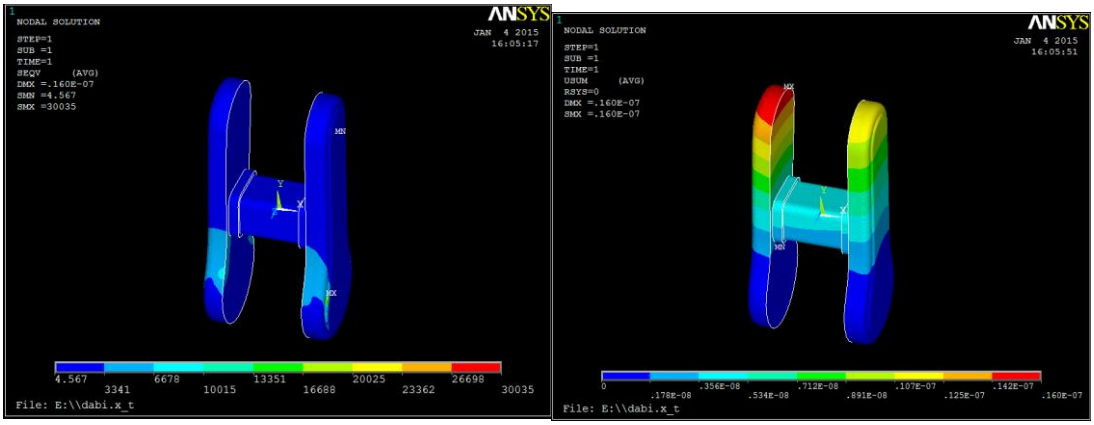


(a) stress nephogram



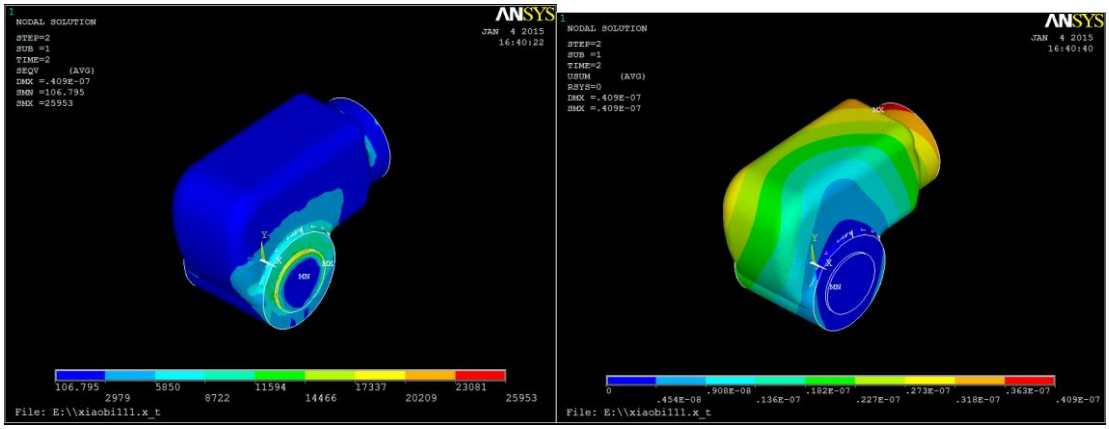
(b) displacement nephogram

Figure 4-10 Statics analysis figure of pillar



(a) stress nephogram (b) displacement nephogram

Figure 4-11 Statics analysis figure of upper arm



(a) stress nephogram (b) displacement nephogram

Figure 4-12 Statics analysis figure of forearm

According to the analysis results, the maximum stress and strain of the four rigid body is shown in table 4-4.

Table 4-4 Statics analysis result

Rigid body / **Results**	Base	Pillar	Upper arm	Forearm
Maximum stress (MPa)	0.0295248	0.00025080	0.0030035	0.0025923
Maximum displacement (mm)	0.0000204	0.00000863	0.0000160	0.0000409

(4) Modal analysis

Similar to the step (3), set up the material elastic modulus to 2.2e11, poisson's ratio to 0.3, the density to 7800, and then analyze it. The four main body structures' first five order natural frequency are onbtained respectively as shown in figure 4-13~4-12, and the first order vibration modal is as shown in figure 4-14.

***** INDEX OF DATA SETS ON RESULTS FILE *****

SET	TIME/FREQ	LOAD STEP	SUBSTEP	CUMULATIVE
1	1498.1	1	1	1
2	1712.0	1	2	2
3	2329.9	1	3	3
4	3104.8	1	4	4
5	6294.5	1	5	5

(a) The first five order frequency of base

***** INDEX OF DATA SETS ON RESULTS FILE *****

SET	TIME/FREQ	LOAD STEP	SUBSTEP	CUMULATIVE
1	1420.2	1	1	1
2	1843.7	1	2	2
3	2484.7	1	3	3
4	4036.2	1	4	4
5	4120.2	1	5	5

(b) The first five order frequency of pillar

***** INDEX OF DATA SETS ON RESULTS FILE *****

SET	TIME/FREQ	LOAD STEP	SUBSTEP	CUMULATIVE
1	1237.8	1	1	1
2	1858.6	1	2	2
3	3137.7	1	3	3
4	3222.5	1	4	4
5	3492.1	1	5	5

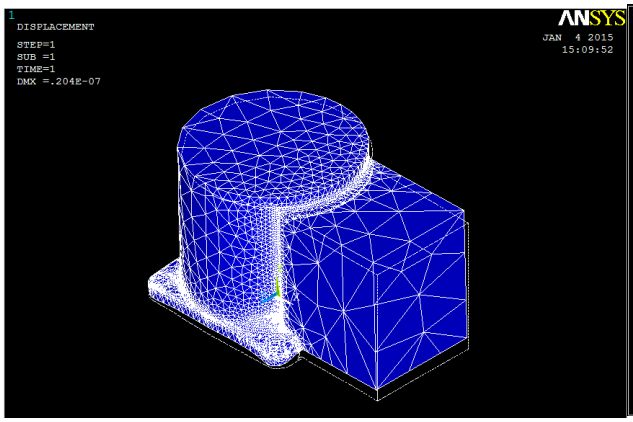
(c) The first five order frequency of upper arm

***** INDEX OF DATA SETS ON RESULTS FILE *****

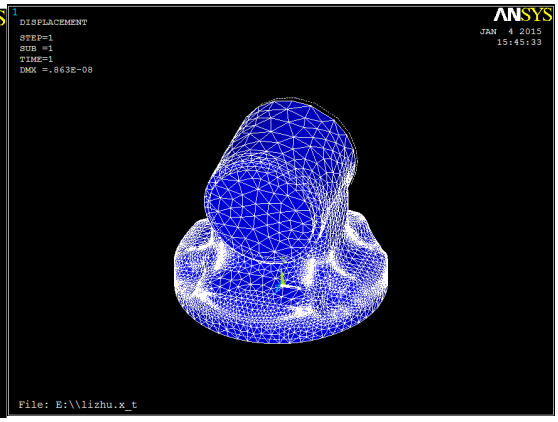
SET	TIME/FREQ	LOAD STEP	SUBSTEP	CUMULATIVE
1	1067.3	1	1	1
2	1986.3	1	2	2
3	2518.2	1	3	3
4	2858.6	1	4	4
5	4025.8	1	5	5

(d) The first five order frequency of forearm

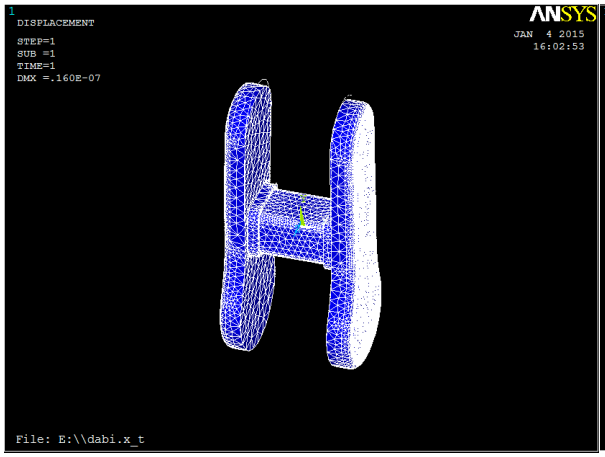
Figure 4-13 The first five order natural frequency of main body structure



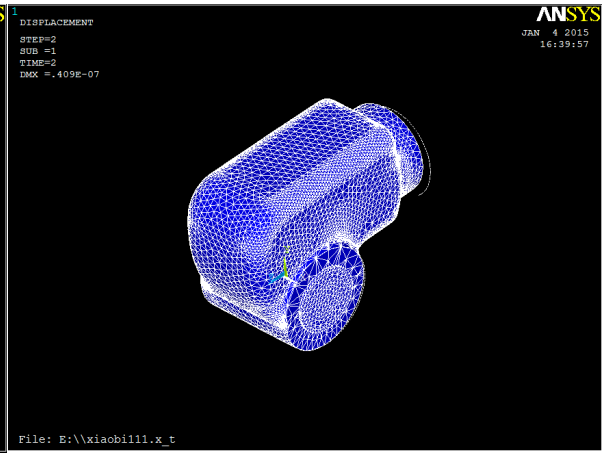
(a) vibration mode of base



(b) vibration mode of pillar



(c) vibration mode of upper arm



(d) vibration mode of forearm

Figure 4-14 The vibration mode of main body structure

4.7 Chapter Summary

In this chapter, firstly the industrial robot user-customized configuration design template is constructed, and users can obtain the industrial robot meeting constraints through parameters setting; Secondly the kinematics and dynamics analysis on template is taken by Simscape model, and the dynamic parameters is analyzed; Thirdly the finite element analysis on template is taken by ANSYS, including statics analysis and modal analysis; Fourthly the parameters flow process in template is analyzed; Finally taking industrial robot user-customized configuration design using configuration template as an example, the configuration template is analyzed in application.

CHAPTER 5 INTERNAL ALGORITHM FOR USER-CUSTOMIZED CONFIGURATION DESIGN

Formal components and configuration template constitute the design platform database, which is the basis for the design platform to support user-customized configuration design. But only database is not enough to support users designing product successfully, there must be internal algorithm to keep the design running smoothly. Under this principle, the internal algorithm is needed to research, including how to process the uncertainty of the users' requirements? How to retrieve service-components meeting constraints from the database? How to instantiate the service-components to objects? How to choose the best objects from a large number of service-components meeting constraints? How to evaluate the configuration results? This chapter describes these issues in depth.

5.1 user-customized configuration design process analysis

In configuration design process, generally the predefined components collection, configuration requirements description and configuration standard are provided by a product configuration system [152][153][154]. Similarly, in user-customized configuration design process, the set of service-components and the configuration constraints are provided by product design platform. The specific configuration design process through configuration system by user is shown in figure 1.

From the horizontal point of view, the configuration process is divided into four levels: The first level is to convert the user's needs to specific configuration parameters. The main contents are the user enters the product requirements and tasks prompted by configuration template, change the undetermined parameters of product configuration template to determined parameters, and then extract parameters needed by Simscape model calculation to input. The Simscape model obtains parameters' values such as joint torque, velocity, etc. These parameters are seen as the basis for service-component configuration; The second level is service-component configuration, the main contents are the retrieving and instantiation of service-component configuration. According to the configuration parameters obtained in first level, the service-component similarity in domain ontology is calculated, and the most similar service-components are selected as part of the template. Then the service-components of template are instantiated to components. These components are retrieved from the component library, and the retrieve results constitute the configuration program meeting configuration parameters and template constraints. When the components meeting constraint does not exist, the service-component will be extended, and the extension begins from the basic value

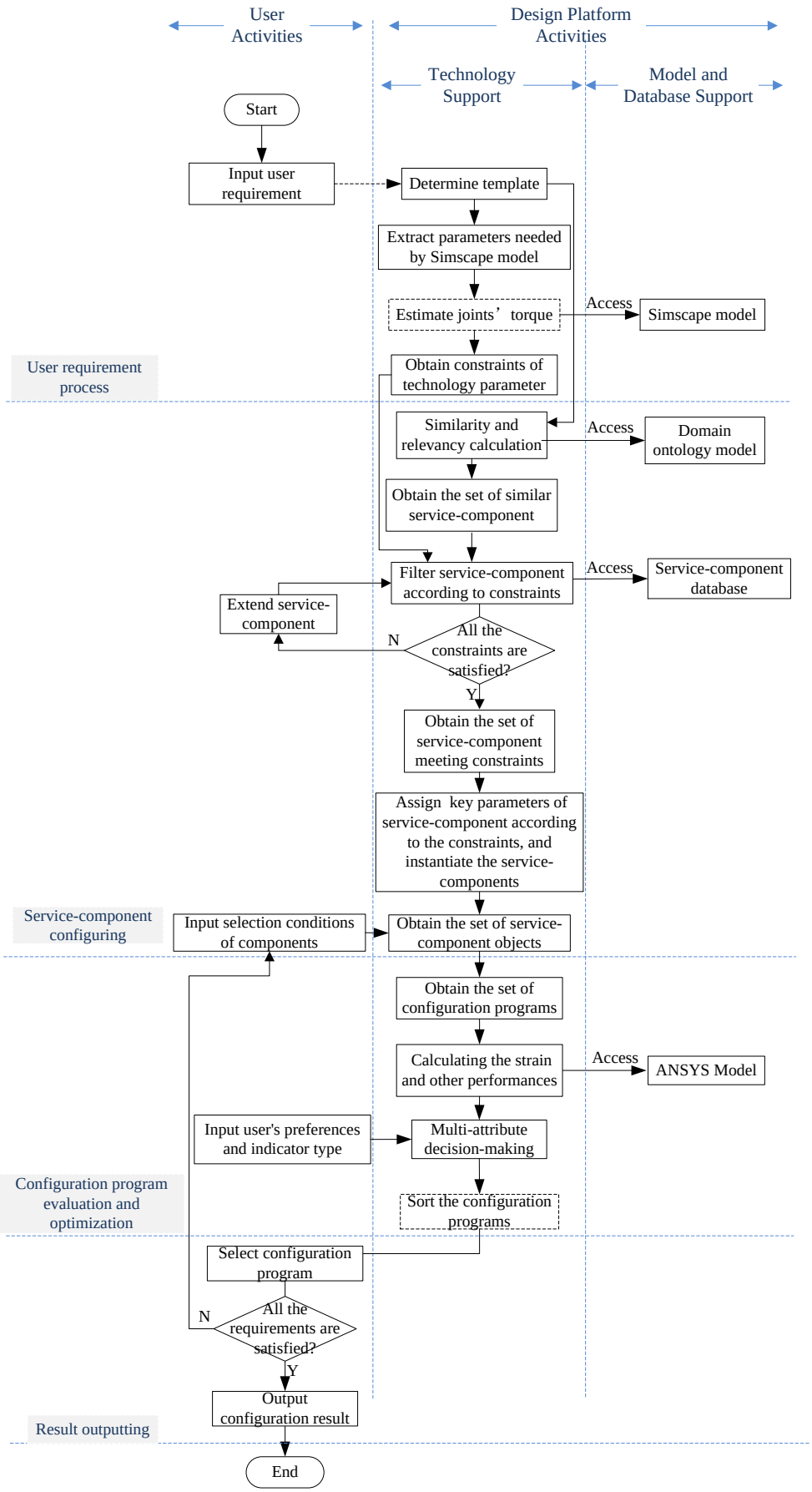


Figure 5-1 The process of user-customized configuration design

expansion; The third level is the configuration program evaluation and optimization. The components meeting constraints constitute the set of configuration program. The product performances of each program are calculated, including stress and strain, vibration response and other parameters of interest to users, such as price, origin, etc. Integrated fuzzy evaluation method is taken to evaluate the configuration program, and the configuration programs are sorted according to the evaluation results; The fourth level is to obtain the final configuration result. According to the sorted result of the previous level, the result in the first place is selected as the final configuration result. As all requirements are verified to meet all requirements, this result is output.

As described in Chapter 2, the design platform is the key to achieve user-customized configuration design of industrial robot. According to Figure 5-1, in order to complete the whole process, the platform should be able to complete all the activities of the second and third columns. The model and database support in the third column has been done in detail in Chapters 3 and 4, and this chapter focuses on the second column.

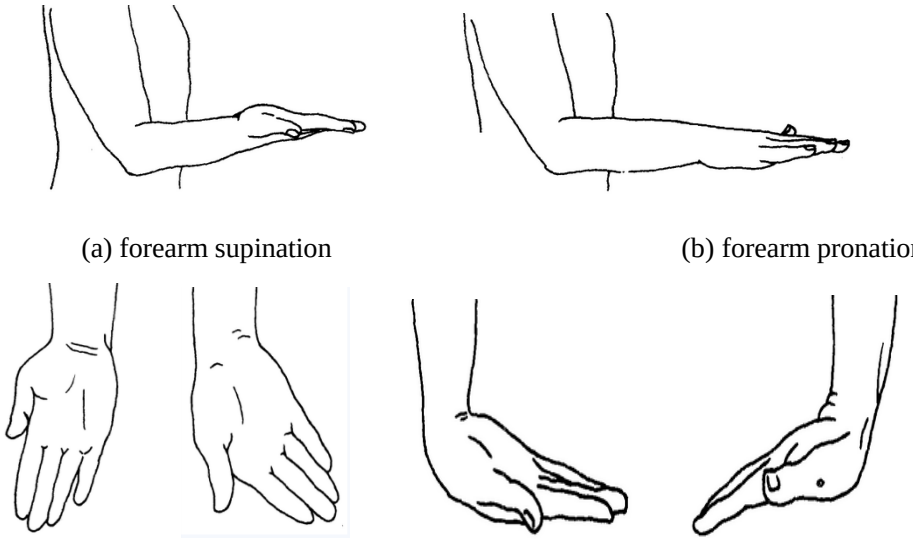
5.2 User requirements processing

5.2.1 Determine the DOF of wrist joint

The purpose of this step is converting the user's needs to parameter constraints for service-components. Firstly user enter parameters mentioned in Chapter 4 such as prime motor type, transmission mode input template. But the number of DOF and the movement type can’t be determined directly. User usually only can give the task of industrial robot to complete, but does not clear how many DOFs are appropriate for this task. This problem requires configuration template to process.

The DOF of wrist joint is designed to mimic the human wrist. The movement that human wrist could be performed is shown in Figure 5-2 [155]. Wherein Figure 5-2(a) and Figure 5-2(b) actually belong to arm activity, but they are designed as rotational degree of freedom for wrist joint in the general design of industrial robot. By this way, when the robot is rotating, there is no need to drive the arm but only drive the wrist and the end effector, which can reduce the load for the robot. Figure 5-2(c) and Figure 5-2(d) are designed to the first rotational degree of freedom of the wrist joint. When the first rotational degree of freedom rotates 90 degrees, the second swing degree of freedom can complete the palmar flexion and dorsiflexion movement in Figure 5-2(e) and Figure 5-2(f), so the Figure 5-2(e) and Figure 5-2(f) can be omitted in robot. Therefore, a rotational degree of freedom and a swing degree of freedom can complete the movement that can be done by human.

Although the wrist joint with two degrees of freedom can mimic the human wrist to complete various sports, but in actual use it will encounter some problems. For example, industrial robots need to complete tasks such as drilling or screwing. If the screw hole location or the center position is coaxial with the first degree of freedom of wrist joint, the robot can easily accomplish this task. But this situation is rare, when the hole is not coaxial with the first degree of freedom of wrist joint, the wrist is difficult to complete the rotating task. In order to increase the flexibility of the wrist, a rotational degree of freedom is usually coupled with the swing degree of freedom. So that when the end effector is required to complete tasks such as drilling, it can be realized through the final rotational degree of freedom, without the need to mobilize the arm joint, wrist joint and other parts to carry out.



(a) forearm supination (b) forearm pronation

(c) radial deviation (d) ulnar deviation (e) palmar flexion （f） dorsiflexion

Figure 5-2 Human wrist movement

Based on the above analysis, the universal wrist structure recommend by configuration template for user is shown in Figure 5-3. The wrist is composed of a degree of freedom of rotating around X-axis, a degree of freedom of swing around Y-axis and the last one degree of freedom of rotating around X-axis. The wrist joint can replace human to complete most of the activities. More degree of freedom could increase flexibility of industrial robot, but it also increases the controlling complexity. Therefore, the configuration template still provide the statement of the number of degrees of freedom and the task can be completed, as shown in Table 5-1. User can select the appropriate number of degrees of freedom based on practical task.

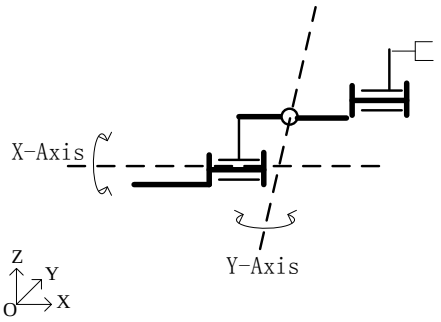
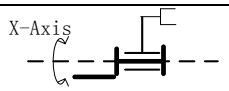
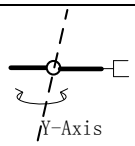
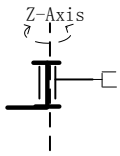
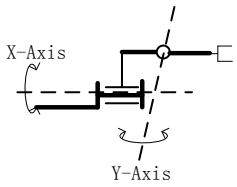


Figure 5-3 Universal wrist structure

Table 5-1 The wrist DOF and movement range

The number of DOF	The kinematic pair type	The movement range
0	无	There is no DOF for the wrist, and the wrist can't move in space.
1		The wrist has a DOF of rotating around X-axis, and can rotate around X-axis.
		The wrist has a DOF of swing around Y-axis, and can pitch around Y-axis.
		The wrist has a DOF of rotating around Z-axis, and can swing side-to-side.
2		The wrist can complete any activities except drilling or screwing.

After user inputting all the parameters, the configuration template will be determined, and the template will calculate the parameters constraints between service-components automatically.

5.2.2 The processing of user fuzzy requirements

In addition to the degrees of freedom, user needs to enter the other performance parameters. In design process, the user's requirements may be not precise, for example, the requirement for arm length may be an accurate numerical value, but it also may be an imprecise value range. The fuzzy set theory is usually used to deal with these imprecise requirements [156][157][158]. The fuzzy requirements are processed as follow in this paper. Set the domain of discourse to X, and x is element in X. Set S as asymmetric triangular fuzzy number, and there are following formulas [157]:

$$S = (S^C, S^S) \tag{5.1}$$

$$S^L = S^C - S^S \tag{5.2}$$

$$S^R = S^C + S^S \tag{5.3}$$

$$\mu_s(x) = \begin{cases} 1 - \left| \dfrac{S^C - x}{S^S} \right| & if\ S^L \le x \le S^R \\ 0 & otherwise \end{cases} \tag{5.4}$$

Where S^C is the center value of S, and S^S is the spread value, $\mu_s(x)$ is the membership function.

In the industrial robot configuration design field, set the parameters set to be determined in the template is $X=\{x_1,x_2,...x_n\}$, n is the number of parameters to be determined in the template. The range of values for each parameter is given by the platform according to their own database, the range of values for the p_{th} parameter x_p is $[x_p^{min}, x_p^{max}]$. There is also a fuzzy numbers set $S=\{S_1, S_2, ...S_n\}$, and S_n represents the value assigned by the user to the parameter to be determined. According to user's demand, each fuzzy number has an expectation value S_i^C, an allowed spread value S_i^S. When the user's demand can be determined precisely, the value of S_i is 0. So, when the value of S_i^S is not 0, the membership function of the p_{th} parameter x_p in template to the p_{th} parameter S_p in fuzzy set S is as follow:

$$\mu_S(x_p) = \begin{cases} 1 - \left| \dfrac{S_P{}^C - x_P}{S_P{}^S} \right| \cdots\cdots if \cdots x_p \in (x_P{}^L, x_P{}^R) \\ 0 \cdots\cdots otherwise \end{cases} \tag{5.5}$$

$$\mu_S(x_p) = \begin{cases} 1 \cdots if \cdots x_p = S_P{}^C \\ 0 \cdots if \cdots x_p \ne S_P{}^C \end{cases} \tag{5.6}$$

Wherein, $x_p{}^L = Max(S_P{}^L, x_p{}^{\min})$, $x_p{}^R = Min(S_P{}^R, x_p{}^{\max})$, $x_p{}^L \le x_p{}^R$

Membership function indicates the degree of parameter satisfaction to the user's demand. The greater the degree of membership, the closer the parameter value to the user's needs. When $x_p=S_P{}^C$, the membership value is 1, indicating that the template parameter value fully meets the user's expectation.

Since the configuration template obtaining the user's fuzzy demand, the value of x_i with the maximum membership to the fuzzy demand is calculated. The value is input to Simscape model as template parameter value to calculate the joint torque and other parameters. These parameters provide the basis for component retrieval as the final configuration parameters.

5.3 Service-component configuration

5.3.1 Similarity and relevancy calculation of service-component

Since the user input requirements to the template, the industrial robot's main structure and service-component type are determined. However, only the main structure is not enough, the standard components connected to these main structure, the related components supporting the main structure, and better cost-effective components than the selected components are also needed. It is possible for the configuration template to include all types of components, which will increase the degree of complexity for the user configuration work. But these types of components are essential for an industrial robot. Therefore, based on ontology similarity and relevancy calculation, this section retrieve service-component after determining indiustrial robot structure. The goodness of fit between calculation results and real situation will depend on the accuracy degree of domain ontology reflecting real-world. So the domain ontology requires constant revision and improvement according to the situation, so that the service-component obtained by calculating will be more accurate.

In the ontology, the concept similarity refers to the extent of interchangeably for two concepts having similarity in function. When calculating the similarity, only the hyponymy relation is considered; the concept relevancy means the probability of appearing at the same time for two concept having dependencies on the function. When calculating the relevancy, all the relations in ontology except for the hyponymy relation are considered. For example, car and bicycle has similarity, but car and gasoline has relevancy [159]. There are many ontology concept similarity and relevancy calculation method, such as domain ontology calculation method based on semantic distance [160][161], domain ontology calculation method based on based on the contents [162], heterogeneous ontology concept mapping method based on semantic features similarity [163][164], mixing calculation method integrated the various methods [165], etc. The similarity influence factors between the two concepts contains the following aspects: the hierarchy of concept in the ontology, the distance between the concepts, attributes similarity between the concepts an so on.

In the existing methods, most methods are concept similarity calculation based on the concept hyponymy relation. Only a few methods consider the other relations, but these relations are considered as a component of similarity, and not be distinguished as relevancy separately. In the retrieval process of service-component, the two service-components with high degree of similarity can interchange, and the two service-components with high degree of relevancy need to be selected simultaneously. The application of similar

service-components and relevant service-components has essential difference, and they can't be seen as integrated similarity. Therefore, in this section the service-component retrieval based on ontology similarity and relevance has two directions: one is the similar service-component retrieval based on the hyponymy relation in ontology; the other one is relevant service-component retrieval based on all the relations in ontology except for the hyponymy relation.

(1) The service-component similarity algorithm based on the hyponymy relation in ontology

Assuming that in ontology only having hyponymy relation, the concept has n-tier, so the edge has *n-1* layer($n \geq 2$). The length of the edge is relevant with the connected concept granularity. The concepts connected by edge is divided finer, the greater the similarity between the two concepts, so the smaller the length of this edge. Assuming that there are two service-components *X* and *Y*, the corresponding concepts in ontology are C_x and C_y. The service-component similarity algorithm based on the hyponymy relation in ontology is shown as formula (5.7) - (5.8):

$$L_{Pi} = \frac{\ln(n-1)+1}{(\ln D_{Pi}+1) * N_{Pi}} \tag{5.7}$$

$$\mathrm{Sim}_{\mathrm{Dis}}(C_x, C_y) = \frac{\alpha}{\sum_{i=1}^{m} L_{Pi} + \alpha} \tag{5.8}$$

Where P_i is the shortest path connecting two concepts, *m* is the total number of the shortest path, L_{Pi} is the edge length of path, D_{pi} is the level of edge P_i in the ontology, N_{Pi} is the number of sub-concepts owned by the upper concept of the two concepts connected by edge P_i. α is an adjustable parameter between 0 and 1. When $\sum_{i=1}^{m} Lp_i = 0$, the distance between the two concepts is 0. So $\mathrm{Sim}_{\mathrm{Dis}}(C_x, C_y)$=1, that the two concepts is the same. When $\sum_{i=1}^{m} Lp_i = +\infty$, there is no path between the two concepts. So $\mathrm{Sim}_{\mathrm{Dis}}(C_x, C_y)$=0, means that there is no resemblance between the two concepts.

In addition to the concept, the service-component also has attributes, that is the *ob* in formula 2.1. The degree of attributes similarity is an important aspect for affecting service-component similarity. Set up the attributes of two service-components *X*, *Y* to ob_x and ob_y respectively. The value range of attributes are ob_x^v and ob_y^v. The mean value of attributes are $ob_x^{\bar{v}}$ and $ob_y^{\bar{v}}$. The common attributes shared by two service-components is ob_{xy}, so $ob_{xy}=\{ob_1, ob_2, \ldots ob_t\}$, wherein *t* is the number of common attributes. The formula for

calculating two service-components similarity based on attributes are as follows:

$$ED = \sqrt{\sum_{i=1}^{t}(ob_{xy}x_i^{\bar{v}} - ob_{xy}y_i^{\bar{v}})^2} \tag{5.9}$$

$$P = \frac{\varphi(ob_{xy})}{\varphi ob_x} + \frac{\varphi(ob_{xy})}{\varphi ob_y} \tag{5.10}$$

$$Sim_{ob}(X, Y) = \frac{\omega_1\beta}{ED + \beta} + \omega_2 P \tag{5.11}$$

ED is the Euclidean distance between the two attribute vector value ranges composed by the common attributes in *X* and *Y*. *ED* represents the true distance between the value ranges of attributes. $ob_{xy}x_i^{\bar{v}}$ is the mean value of attributes belongs to ob_x and ob_{xy} simultaneously. p $ob_{xy}y_i^{\bar{v}}$ is the mean value of attributes belongs to ob_y and ob_{xy} simultaneously. $\varphi(ob_{xy})$ is the attribute number in ob_{xy}. φob_x (non-zero) is the attribute number in ob_x. φob_y (non-zero) is the attribute number in ob_y. *P* is the number similarity of the same attribute in two service-components. Sim_{ob} is attribute similarity of service-components *X* and *Y*. ω_1 and ω_2 is the weight for *ED* and *P* in attribute similarity. β is an adjustable parameter.

The comprehensive similarity of two service-components is calculated as follows:

$$Sim(X, Y) = \lambda_1 \mathrm{Sim}_{\mathrm{Dis}}(\mathrm{C_x}, \mathrm{C_y}) + \lambda_2 Sim_{ob}(X, Y) \tag{5.12}$$

λ_1 and λ_2 are the weight for similarity value based on distance and similarity value based on attribute in service-component similarity calculation.

(2) The service-component relevency algorithm based on all the relations in ontology except for the hyponymy relation

The domain ontology constructed in the article has seven horizontal relations. In which there are three relations express relations between different concepts: *Ha*, *Ca*, *As*. The relevency of concepts connected by different types of horizontal relations is different. The strength order of the association of three relations is *Ca*> *As*> *Ha*. For example, the relevency of two concepts connected by *Ca* relation is greater than the relevency of two concepts connected by *Ha* relation. Asuming that there are two service-components *X* and *Y*, the relevency algorithm of the two service-components is as follows:

$$L_{Ca} = \omega_1 \tag{5.13}$$

$$L_{As} = \omega_2 \tag{5.14}$$

$$L_{Ha} = \omega_3 \tag{5.15}$$

$$L_{Pi} = k_i L_{Ca} + m_i L_{As} + n_i L_{Ha} \tag{5.16}$$

$$\mathrm{Re}\ I(X,Y) = \frac{\delta}{\sum_{i=1}^{t} L_{Pi} + \delta} \tag{5.17}$$

$\omega_1\ \omega_2\ \omega_3$ is the weights of three relations, there is $\omega_1 > \omega_2 > \omega_3$. P is the three relations' set of all paths between X and Y. t is the number of path in the set, P_i is the i*th* path in the set. k_i is the number of relation *Ca* in P_i. m_i is the number of relation *As* in P_i. n_i is the number of relation *Ha* in P_i. L_{Pi} is the length of P_i. δ is adjustable parameter.

Using the relevency algorithm, the service-components set relevant with main structure of industrial robot can be obtained. But the service-components in this set are repetitive, there are two reason for the repetition: one is that if service-component *A* and service-component *B* is connected to each other, the service-components set relevant with *A* must caontains *B*, and the service-components set relevant with *B* must caontains *A*; another one is that if *A* and *B* is connected through middle component *C*, the service-components set relevant with *A* and the service-components set relevant with *B* will both contain *C*. Therefore, the set will be reduction after relevency calculation. Set the service-components set of industrial robot's main structure to *M*. The middle components set connecting two service-components in *M* but not belonging to *M* is *I*. The relevant service-components set of service-components in *M* is M_{Rel}. Then the relevant service-components set after reduction is $M'_{Rel}=M_{Rel}-M-I$.

5.3.2 Service-component selection and instantiation

Before service-component instantiation, the service-component should be selected by technical parameters constraints obtained in section 5.2. The service-components whose value range don't meet the constraints will be taken out of the similar service-components set, and then the remaining service-components will be instantiated. Before instantiation, the instantiation order should be determined first according to the degree of importance of the service-components. The occurrence frequency of service-component in the template internal constraints determines its importance. If one service-component appears frequently, it means that there are many service-components are reciprocal constraint with this service-component. Then this service-component need to be instantiated preferentially, because the parameter value changing of this service-component has a greater impact on other service-components. Learning from the bubble sort method of computer science, the service-components are pairwise compared, and then the instantiation order is determined finally. The service-components set meeting constraints is set to $A=\{S_1,S_2,\ldots S_n\}$, then the algorithm to determine instantiation order of service-components is as follows:

a) Reduction the set. Traverse the elements in set of *A*. If S_i and S_j is similar

service-components which obtained by similarity calculating, then one of them is remained and another one is deleted;

b) Take any two adjacent service-components S_j and S_{j+1} from set A, then count the number of the occurrence frequency of service-component in the template internal constraints, which recorded as $N.S_j$ and $N.S_{j+1}$. If $N.S_j < N.S_{j+1}$, then exchange the place;

c) When the number of the occurrence frequency of the adjacent two service-components is the same, the two service-components will be placed in one bracket as one element in set A;

d) Repeat step *a)* and *b)* for each pair of adjacent service-components in set A. The operation starts with S_1 and S_2, and end with S_{n-1} and S_n;

e) Repeat step *a)* and *b)* until all the elements in set A meeting $N.S_j > N.S_{j+1}$;

f) Add the deleted elements of step *a)* into set A again, then put them in one bracket with similar elements as one element in set A;

g) Output the ordered service-component set $A' = \{S_{1'}, S_{2'}, \ldots S_{n'}\}$, wherein $n' \leq n$.

After determining the instantiation order of service-components, the service-component in set A' is instantiated from the first element. There is no need to assign value to every parameter of service-component. Because the value range of every parameter of service-component is composed by the service-component objects. When one or several parameters of one service-component is determined, the service-component object will be determined, so the instantiation is completed. Assignment to parameters depends on the technical constraints output by configuration template. According to the constraint boundary, the value nearest boundary constraint is selected from the value range as a parameter value of service-component, so as to determine the service-component object.

5.3.3 Service-component object selection

The service-component is instantiated from the technical perspective in Section 5.2.2. But in the service-component database, there may be more than one service-component object with same technical parameter value. For example, the same type of component may be uploaded to the database by two different manufacturers, so there are two different service-component objects with same technical parameter value in the database. Therefore, the service-component objects meeting technical parameter requirements need to be further selected.

The selection process is based on the value of *Ma* in service-component. The user set priority for various parameters in *Ma* according to their needs. The main purpose of selection is that the system can choose the most appropriate service-component object accordance with user preference when there are more than one service-component objects with the same technical parameter value. The parameters concerned by user in *Ma* contains the information released by the manufacturer such as component suppliers, price, place of origin, ease of use, adequacy of supply, delivery and other information. Assuming there are two service-component objects having the same technical parameter value. If the object with lower price should be selected preferentially in terms of user intent, then the price parameter is set to the highest priority. If the object with adequate supply should be selected preferentially in terms of user intent, then the supply is set to the highest priority. The parameters of *Ma* is recorded as selection vector $PU=\{pu_1,pu_2,\ldots pu_n\}$. The selection vector by user reordered is recorded as $PU'=\{pu_{1'},pu_{2'},\ldots pu_{n'}\}$, wherein the priority of pu_i is greater than pu_{i+1}. PU' is the selective basis of service-component object. The selected service-component object can form the configuration program set for further use.

Through the above analysis, it can be seen that the configuration process is closely related to the service-component representation method. Figure 5-4 shows how the representation method to play a role in the configuration process.

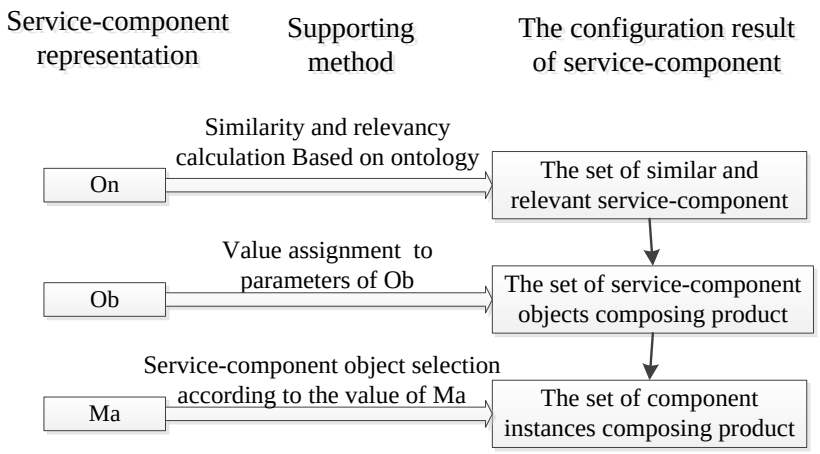


Figure 5-4 the supporting of representation method to configuration process

5.4 Configuration program evaluation

Every program of configuration program set can output program performance parameters. The parameters contain parameters output through the configuration template such as workspace, work velocity, price, etc., and also contain parameters obtained by finite element analysis such as vibration response, the stress and strain value, etc.. The program in line with user preferences is selected according to evaluating the performance of different programs.

Each configuration program has multiple performance parameters, and a comprehensive evaluation is needed to make by considering the multiple performance parameters. Because of the uncertainty of performance evaluation, evaluation methods commonly used contains fuzzy comprehensive evaluation method, fuzzy multi-attribute decision making method, and so on. In these methods, every performance indicator of configuration program needs to be graded by the membership function, and the membership function are generally given in advance by the system based on expert experiences. But expert experiences do not represent the user's intention, and the score given by expert to one indicator may not be consistent with the user. Under the user-customized configuration design pattern, the evaluation on configuration program is completely user-oriented. Based on this consideration, the simple linearity weighted multi-attribute decision making method is used to evaluate the configuration program.

a) Decision matrix building

The completed configuration program set is $X=\{x_1,x_2,\ldots x_m\}$. Performance indicator set of program x_i is $Y=\{y_1,y_2,\ldots y_n\}$. Then the decision matrix of configuration program is as follows:

$$R = (r_{ij})_{m\times n} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{5.18}$$

b) Matrix normalized processing

By matrix normalized processing, the dimensional indicator will become dimensionless indicator. The indicator type needs to be divided when processing. In multi-attribute decision theory, the indicator type commonly includes cost type, benefit type, fixed type, deviation type, interval type, deviation interval type, etc.. The indicator types used in this article include the following four types: one is benefit type. The indicator value of this type is positive correlation with the final evaluation results, such as positioning accuracy, the larger the indicator value, the better the overall performance of the program; one is the cost type. The indicator value of this type is negatively correlated with the final evaluation results, such as strain value, the larger the indicator value, the worse the overall performance of the program; the third is fixed type. There is no linear correlation for the indicator value of this type with the final evaluation results. The indicator value of this type is specified by user as the best configuration value, such as the workspace. The user may specify a specific value as the best workspace. The closer the workspace to this value, the better the performance of the program;

the fourth is interval type. The indicator value range of this type is specified by user as the best configuration value range, such as price. The closer the program price to this value range, the better the performance of the program. The type of each indicator belongs to is uncertain, and it is specified by the user according to their needs. For example, the price indicator, if user thinks the lower the price the better, the price will be designated as a cost type. If user feels that the best price is in a certain range rather than the lower the better, the price will be designated as the interval type. Standard 0-1 transformation is used to process the four types of indicator as follows [166]:

$$r_{ij}' = \frac{r_{ij} - Min\ r_j}{Max\ r_j - Min\ r_j} \quad \text{if indicator } r_j \text{ is benefit type} \tag{5.19}$$

$$r_{ij}' = \frac{Max\ r_j - r_{ij}}{Max\ r_j - Min\ r_j} \quad \text{if indicator } r_j \text{ is cost type} \tag{5.20}$$

$$r_{ij}' = 1 - \frac{\left|r_{ij} - r_{0j}\right|}{Max\left|r_j - r_{oj}\right|} \quad \text{if indicator } r_j \text{ is fixed type} \tag{5.21}$$

$$r_{ij}' = \begin{cases} 1 & if\ \ r_{ij} \in \left[r_{0j\min}, r_{0j\max}\right] \\ 1 - \dfrac{Max(r_{oj\min} - r_{ij}, r_{ij} - r_{oj\max})}{Max(r_{oj\min} - Min\ r_j, Max\ r_j - r_{0j\max})} & if\ \ r_{ij} \notin \left[r_{0j\min}, r_{0j\max}\right] \end{cases}$$

$$\text{if indicator rj is interval type} \tag{5.22}$$

The processed decision matrix is as follows:

$$R' = (r_{ij}')_{m\times n} = \begin{bmatrix} r_{11}' & r_{12}' & \cdots & r_{1n}' \\ r_{21}' & r_{22}' & \cdots & r_{2n}' \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1}' & r_{m2}' & \cdots & r_{mn}' \end{bmatrix} \tag{5.23}$$

Wherein $0 \le r_{ij}' \le 1$

c) The index weight determination according to user's preferences

The importance of each indicator is divided into six grades, which is represented by the numbers *0 ~ 5*. The number *0* represents the user believes that the indicator is not important. The number *5* represents the user believes that the importance of this indicator is greatest. The table to obtain the user's preferences is as follows:

Table 5-2 User's preferences grade table

y_1	0	1	2	3	4	5	The recommendation for evaluation
y_2	0	1	2	3	4	5	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
y_n	0	1	2	3	4	5	

The recommendation for evaluation in the rightmost is the reference suggestions provided by system to evaluation program. It can help user to evaluation program more reasonably. After assignment to each indicator, the user's preferences U is obtained, and $U=\{\mu_1, \mu_2, \ldots \mu_n\}$. The U is processed as follows:

$$\mu_i^{'} = \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \tag{5.24}$$

The processed user's preferences $U^{'}=\{\mu_1^{'}, \mu_2^{'}, \ldots \mu_n^{'}\}$, and there is $\sum_{i=1}^{n} \mu_i^{'} = 1$.

d) Comprehensive evaluation value calculation

The comprehensive evaluation value calculation method of configuration program is as follows:

$$F=R^{'} \times U^{'T}=\{f_1,f_2,\ldots f_m\}^T \tag{5.25}$$

Wherein f_i is the comprehensive evaluation value of the i*th* configuration program. If f_i is the maximum value in F, the i*th* configuration program is the best program meeting user's preferences.

The evaluation method takes into account the user's preferences. For the same configuration program, the comprehensive evaluation value obtained by different users is different. In this method, the user only needs to specify the performance indicator type and grade according to their needs, then the system can automatically calculate the final score for each configuration program to provide a reference for the user.

5.5 Case study for user-customized configuration design

a) User's requirements processing

According to the parameter requirements of configuration template in section 4.2.2, the parameters to be determined contain static parameters $P_s=\{N_{DOF}, T_{KP}, T_{DS}, T_{Tr}, T_{Se}, T_{EE}, W_{EE}, L_a\}$, and dynamic parameters $P_d=\{\omega_p, A_p, \omega_{ee}, A_{ee}\}$. For the parameters N_{DOF} and T_{KP}, according to the method in Section 5.2.1, the user selects the wrist around the X and Y axis.

Together with the position degree of freedom, the final industrial robot has five degrees of freedom. Then the user determines the type of prime mover is motor drive, the transmission mode is gear reducer. According to the actual task to be completed, the industrial robot needs carry plates, so the negative pressure type sucker appropriate to carry plates is selected as the end effector. The maximum weight of plates that is the maximum load on end effector is 4kg. A force sensor is added to the sucker to measure the pressure. According to the actual using environment, the minimum operating range of industrial robots are required to reach 1.0m, and the maximum operating range should not exceed 1.5m. The velocity of point P is required to reach 2m/s, and the robot can reach this velocity within 0.5s. Thus, the parameters needed to be input by user in configuration template are all determined, and VP_s*={5, (Z,Y,Y,X,Y),* motor, gear reducer, force sensor, negative pressure type sucker, *4kg, (1.0m~1.5m)}*, VP_d*={2m/s, 4m/s*^{2}*, 2m/s, 4m/s*^{2} *}*. Wherein, T_{KP} is a set of values, which records the rotation axis of each degree of freedom. The value of *La* is a fuzzy value. The value having the maximum degree of membership to this fuzzy value is 1.25m, which is calculated by formula (5.5) in Section 5.2.2. Then the fuzzy value of VP_s is replaced by the calculated value.

After all parameters are input, the configuration template becomes a fixed structure in line with the user's needs from an undetermined structure. *M* is the service-component set composing the fixed structure, and *M*={base of *X* type, pillar of *Y* type, upper arm of *Z* type, forearm of *L* type, negative pressure type sucker of *M* series, waist joint (motor of *A* series *1，actuator of *A* series *1， *RV* gear reducer of *B* type *1), shoulder joint (motor of *C* series *1, actuator of *C* series *1, *RV* gear reducer of *D* type *1) , elbow joint (motor of *C* series *1, actuator of *C* series *1, *RV* gear reducer of *D* type *1), wrist joint (motor of *K* series *1, actuator of *K* series *1, harmonic gear reducer of *T* type *1, force sensor of *G* type); at the same time kinematics and dynamics parameters are calculated through the template internal constraints in Chapter 4. By this way, the template transforms the user's needs to the technical parameter constraints of service-components in template to further use.

b) Service-component configuration

Using formula (5.7) - (5.17), the ontology-based similarity and relevance calculation of service-component in *M* is taken. Take the threshold value of similarity and relevance to *0.5*. So the service-components whose similarity and relevance is greater than *0.5* will be output as the results, and the similar service-component set M_{Sim} and relevant service-component set M'_{Rel} are shown in table 5-3.

Table 5-3 The result of similarity and relevance calculation

M	M_{Sim}	M'_{Rel}
Base of X type	Base of X' type, base of X'' type	Joint bolt, connecting shaft, batter board
Pillar of Y type	Pillar of Y' type, pillar of Y'' type	Crossed roller bearing, joint bolt
Upper arm of Z type	Upper arm of Z' type, upper arm of Z'' type	Four-point contact ball bearing, joint bolt
Forearm of L type	Forearm of L' type, Forearm of L'' type	Four-point contact ball bearing, joint bolt
Negative pressure type sucker of M series	Negative pressure type sucker of M' series, negative pressure type sucker of M'' series	Joint bolt
Motor of A series	Motor of A' series, motor of A'' series	Shaft coupling, stand
Motor of C series	Motor of C' series, motor of C'' series	Shaft coupling, stand
Motor of K series	Motor of K' series, motor of K'' series	Shaft coupling, stand
Actuator of A series	Actuator of A' series, actuator of A'' series	Cable
Actuator of C series	Actuator of C' series, actuator of C'' series	Cable
Actuator of K series	Actuator of K' series, actuator of K'' series	Cable
RV gear reducer of B type	RV gear reducer of B' type, RV gear reducer of B'' type	Spring, chain wheel
RV gear reducer of D type	RV gear reducer of D' type, RV gear reducer of D'' type	Spring, chain wheel
Harmonic gear reducer of T type	Harmonic gear reducer of T' type, harmonic gear reducer of T'' type	Spring, belt wheel
Force sensor of G type	Force sensor of G' type	Bolt

Then according to the technical parameter constraints calculated in previous step including base height <0.4m, 0.1m <arm length< 0.5m, motor power> 10W, reduction gear ratio> 20 and other constraints, the similar service-component set is screened. The service-component not satisfying the constraints is deleted, and then the remained service-components are sorted according to the algorithm described in section 5.3.2. he service-component set after being screened and sorted is shown in table 5-4.

Table 5-4 Service-components list after being screened and sorted

M	M_{Sim}	顺序
Base of X type	Base of X'' type	4
Pillar of Y type	Pillar of Y' type, pillar of Y'' type	5
Upper arm of Z type	Upper arm of Z' type	3
Forearm of L type	Forearm of L' type, Forearm of L'' type	3
Negative pressure type sucker of M series	Negative pressure type sucker of M' series	7
Motor of A series	Motor of A' series, motor of A'' series	1
Motor of C series	Motor of C' series	1
Motor of K series	Motor of K' series, motor of K'' series	1

Continued from Table 5-4 Service-components list after being screened and sorted

M	M_{Sim}	顺序
Actuator of A series	Actuator of A" series	6
Actuator of C series	Actuator of C' series, actuator of C" series	6
Actuator of K series	Actuator of K' series	6
RV gear reducer of B type	RV gear reducer of B' type, RV gear reducer of B" type	2
RV gear reducer of D type	RV gear reducer of D" type	2
Harmonic gear reducer of T type	Harmonic gear reducer of T' type, harmonic gear reducer of T" type	2
Force sensor of G type	Force sensor of G' type	8

Then the service-components in Table 5-3 are instantiated in order, the service-component object closest to boundary constraints are selected from database as instantiation result from the database. It can be seen that the relevant service-components don't be instantiated here. That is because the relevant service-components are generally standard components, and these components can be selected from database automatically by matching interface information of service-component objects after all the main structure components instantiation is completed.

Finally, the service-component objects are screened. Suppose the parameters order of *Ma* is {low price, short delivery time, ease of use, the origin near to shipping address, adequate supply}. Then for the service-components with the same technical parameters' values, the system will select the service-component of lowest price. If the price is also the same, then the service-component with shortest delivery time is chosen, and so on. After this step is completed, every service-component in Table 5-3 has only one instantiated service-component object for subsequent use.

c) Configuration program evaluation

According to Table 5-3, the number of configuration programs composed by instantiated service-component objects is 2*3*2*3*2*3*2*3=1269. Wherein the numbers of 2,3,2,3,2,3,2,3 are corresponded to the service-component object number of the first eight rows in table 5-3. After the service-component objects of the first eight rows are determined, the following service-component objects can be determined uniquely. For example, when the motor type is determined, the actuator type will be determined. By performance simulation, the computer can get the performance parameters of configuration program. Due to limited

space, only three programs are selected for evaluation here. Assuming the final configuration program set $X=\{x_1,x_2,x_3\}$. The performance indicator set for industrial robots is Y={ maximum working range *(m)*, the maximum load *(kg)*, the maximum velocity *(m/s)*, the maximum strain *(mm)*, the natural frequency *(Hz)*, price (million), weight *(kg)*}. The decision matrix R of configuration program is obtained by formula (5.17) as follows:

$$R=\begin{bmatrix} 1.2 & 5 & 3 & 0.005 & 32.175 & 15 & 137 \\ 1.4 & 4.5 & 2.5 & 0.01 & 51.233 & 12 & 151 \\ 1.25 & 6 & 2 & 0.003 & 48.265 & 16 & 103 \end{bmatrix}$$

The types of performance indicators entered by user are: the maximum working range (interval typr, 1.0m~1.5m), the maximum load (fixed type, 4kg), the maximum velocity (benefit type), the maximum strain (cost type), the natural frequency (benefit type), price (interval type, 10 million~20 million), weight (cost type). Then, according to formula (5.17) - (5.20), the processed decision matrix $R^{'}$ are as follows:

$$R^{'}=\begin{bmatrix} 1 & 0.5 & 1 & 0.714 & 0 & 1 & 0.291 \\ 1 & 0.75 & 0.5 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0.844 & 1 & 1 \end{bmatrix}$$

The importance grade *U* of every indicator is entered by user, and *U={4, 5, 3, 2, 4, 3, 1}*. The processed user preference vector $U^{'}$ is obtained by formula (5.22), and $U^{'}=\{0.18,0.23,0.14,0.09,0.18,0.14,0.04\}$. The comprehensive evaluation result *F* is obtained by formula (5.23):

$$F=\begin{bmatrix} 1 & 0.5 & 1 & 0.714 & 0 & 1 & 0.291 \\ 1 & 0.75 & 0.5 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0.844 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.18 \\ 0.23 \\ 0.14 \\ 0.09 \\ 0.18 \\ 0.14 \\ 0.04 \end{bmatrix} = \begin{bmatrix} 0.651 \\ 0.743 \\ 0.802 \end{bmatrix}$$

As can be seen from the result, the relation of three programs' comprehensive evaluation value is $x_1<x_2<x_3$, and the third program is the configuration result that best meeting the user's preference.

5.6 Chapter summary

This chapter firstly constructs platform-based and user-leading user-customized configuration design process; then the internal algorithm to keep the design running smoothly is studied, including the degree of freedom determination, fuzzy demand calculation, and

service-component configuration. The service-component configuration contains service-component retrieval based on similarity and relevancy calculation by *On*, service-component selection and instantiation by *Ob*, and service-component object selection by *Ma*. The simple linearity weighted multi-attribute decision making method is used to evaluate the configuration program, then the product that best meeting the user's preference is obtained. Finally a case analysis is taken for the internal algorithm

CHAPTER 6 DESIGN PLATFORM PROTOTYPE SYSTEM DESIGN AND APPLIED RESEARCH

In the user-customized configuration design pattern, design platform is an interactive media of all stakeholders. The stakeholders take activities such as design, marketing, purchasing, service and supervision on the platform. Rational design for the design platform is the key to successful implementation of user-customized configuration design pattern.

6.1 System Requirements Analysis

6.1.1 Use case diagram for platform

Figure 6-1 is a use case diagram of the platform, which shows the different stakeholders' activities on the platform in detail. When the customers only design product but not purchase product on the platform, they actually act as a product designer, as definition 4 in Chapter 2, these two roles are collectively called user.

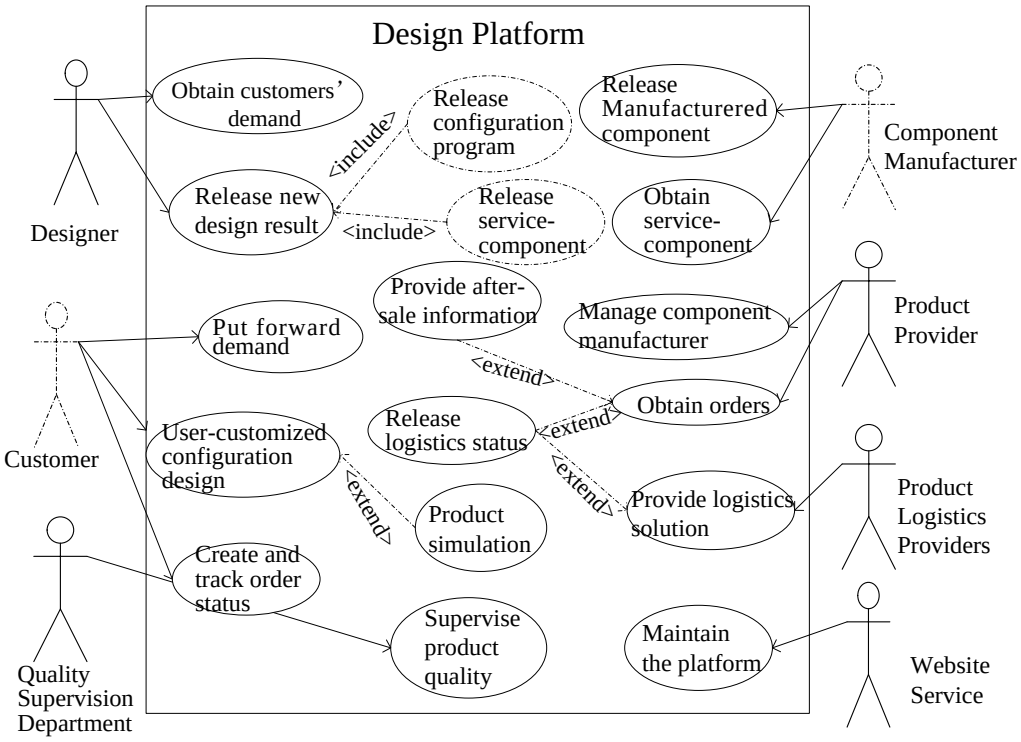


Figure 6-1 Use case diagram of platform

Designers receive customers' needs and release new design result. Component manufacturers obtain service-components and release produced components. Users design the product, and as customer, they can also put forward new demand and track their order after purchasing product. Product providers manage components manufacturers, obtain product orders and provide after-sale information to customers. Product logistics providers provide the

logistics solution, and release the order status. Quality supervision departments supervise the various product and service quality of the participants. Website service providers provide background maintenance, and resolve the technical problems encountered by the participants in the process of using platform.

6.1.2 Role-based business process analysis

As shown in the use case diagram, there are seven kinds of platform roles. Figure 6-2 is the business process diagram for customers using the platform. When customers are ready to buy industrial robot, they have three options: one is to search the product directly on the platform. If the database is not able to meet their requirements, customers can design product by themselves, and also can release requirements on the platform to wait solutions of other designers.

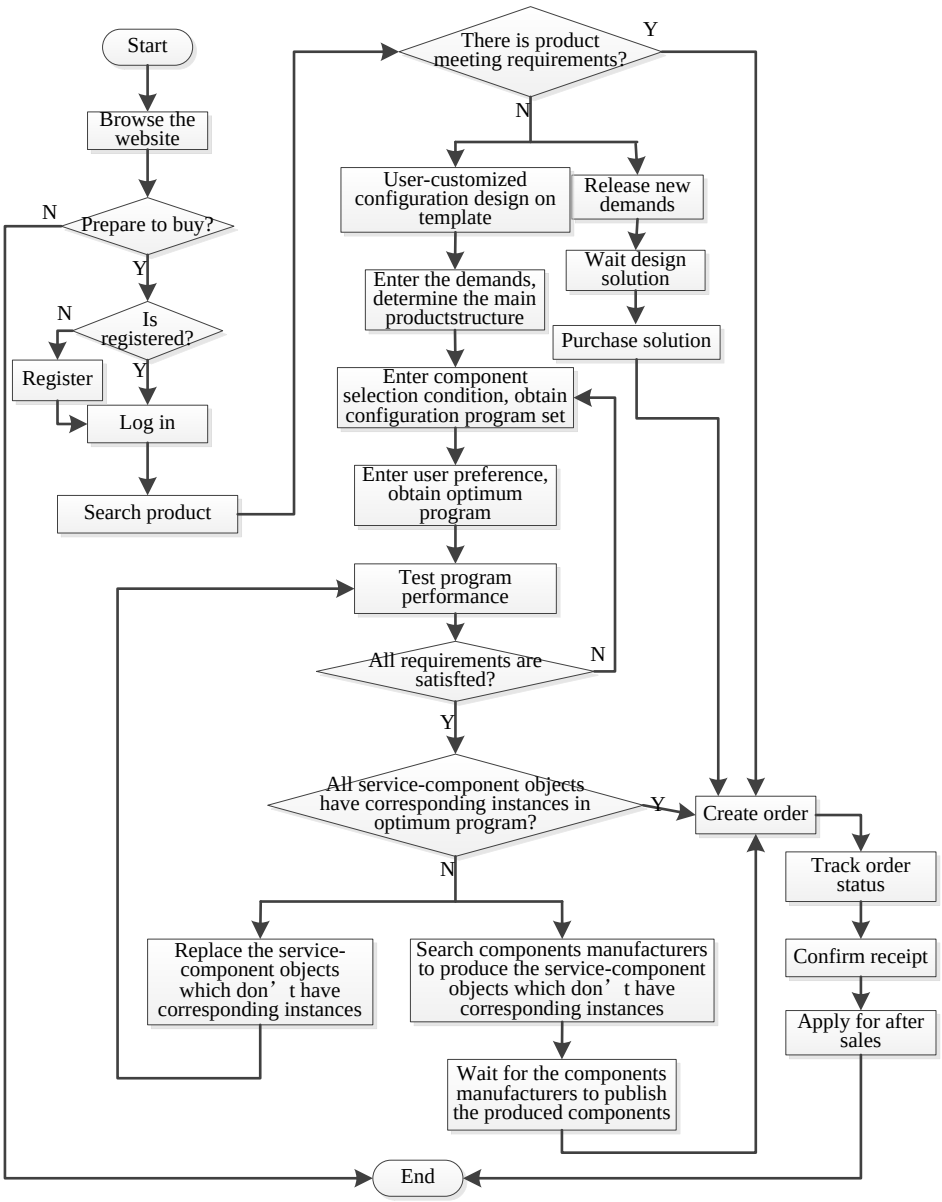


Figure 6-2 The business process diagram for customers

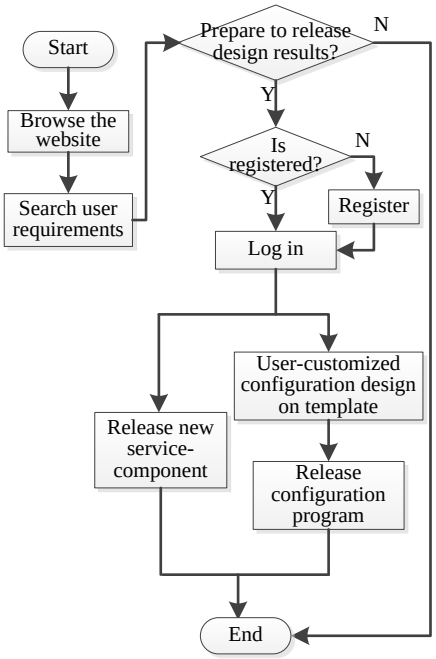


Figure 6-3 The business process diagram for designers

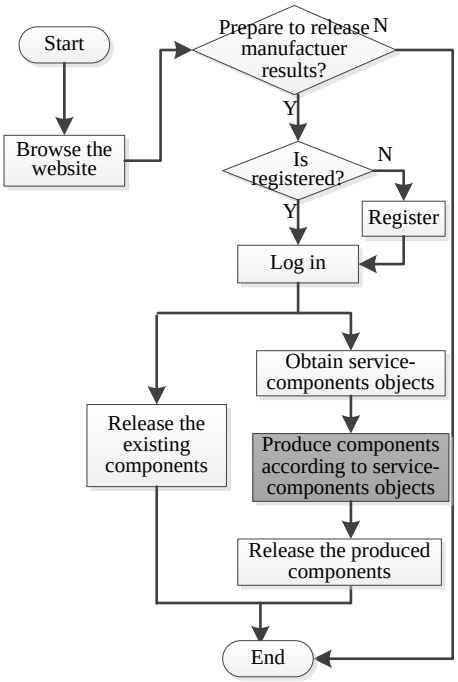


Figure 6-4 The business process diagram for component manufacturers

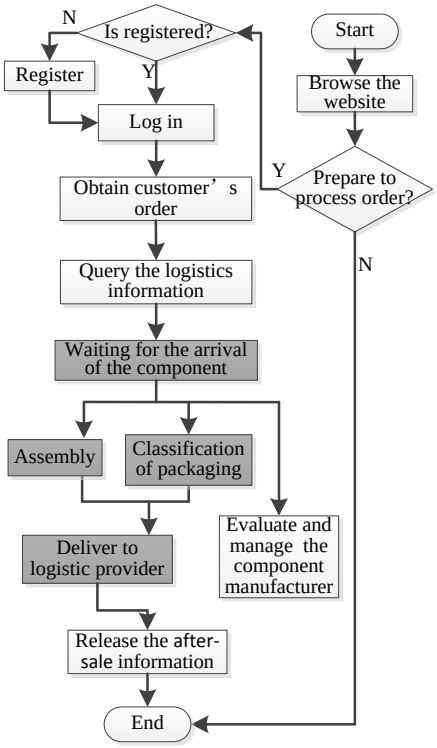


Figure 6-5 the business process diagram for product providers

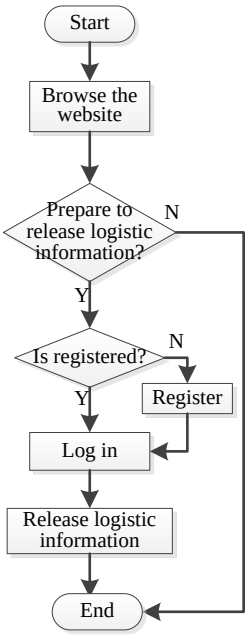


Figure 6-6 the business process diagram for product logistics providers

Figure 6-3 is the business process diagram for designers using the platform. Wherein the

detailed process of "User-customized configuration design on template" is similar to the process in Figure 6-2, it is no longer carried out in detail. Figure 6-4 is the business process diagram for component manufacturers using the platform. Component manufacturers can publish their own components directly to the platform, and also can select the service-components designed by other designers from the platform to manufacture. Figure 6-5 is the business process diagram for product providers using the platform. Products providers are mainly for product assembly and packaging offline, and then publish product after-sale service information on the platform. Figure 6-6 is the business process diagram for product logistics providers using the platform. The deep boxes in process diagram are the offline activities undertaken by the participants, and not the part of the business activities on the platform.

6.2 System design

6.2.1 Function module design

According to requirements analysis, design platform is divided into seven modules: product display, user-customized configuration design, product release, product trade, logistics management, user management, back-stage management. Functional block diagram of the platform is shown in Figure 6-7. The contents in deep boxes are the core of product design platform, and are the parts with core competitiveness of the design platform compared to the general e-commerce platform. The contents in light boxes are similar to the contents of general e-commerce platform, and not emphasis here.

a) Product release. This section includes two contents: designers release their design results and component manufacturers release their manufacture results. Designers release service-components to service-components database for user-customized configuration design invoking. Designers also can release configuration program via user-customized configuration design; component manufacturers will release the manufactured components and industrial robots to platform as service-component instances for customers to buy.

b) Product display. This section displays the contents released by product release module, including design results such as service-component objects and configuration programs, manufacturing results such as components, industrial robots. Platform user can select the products and service-components according to different conditions, such as according to price, according to sales volume, according to a technology parameter, according to task, and so on.

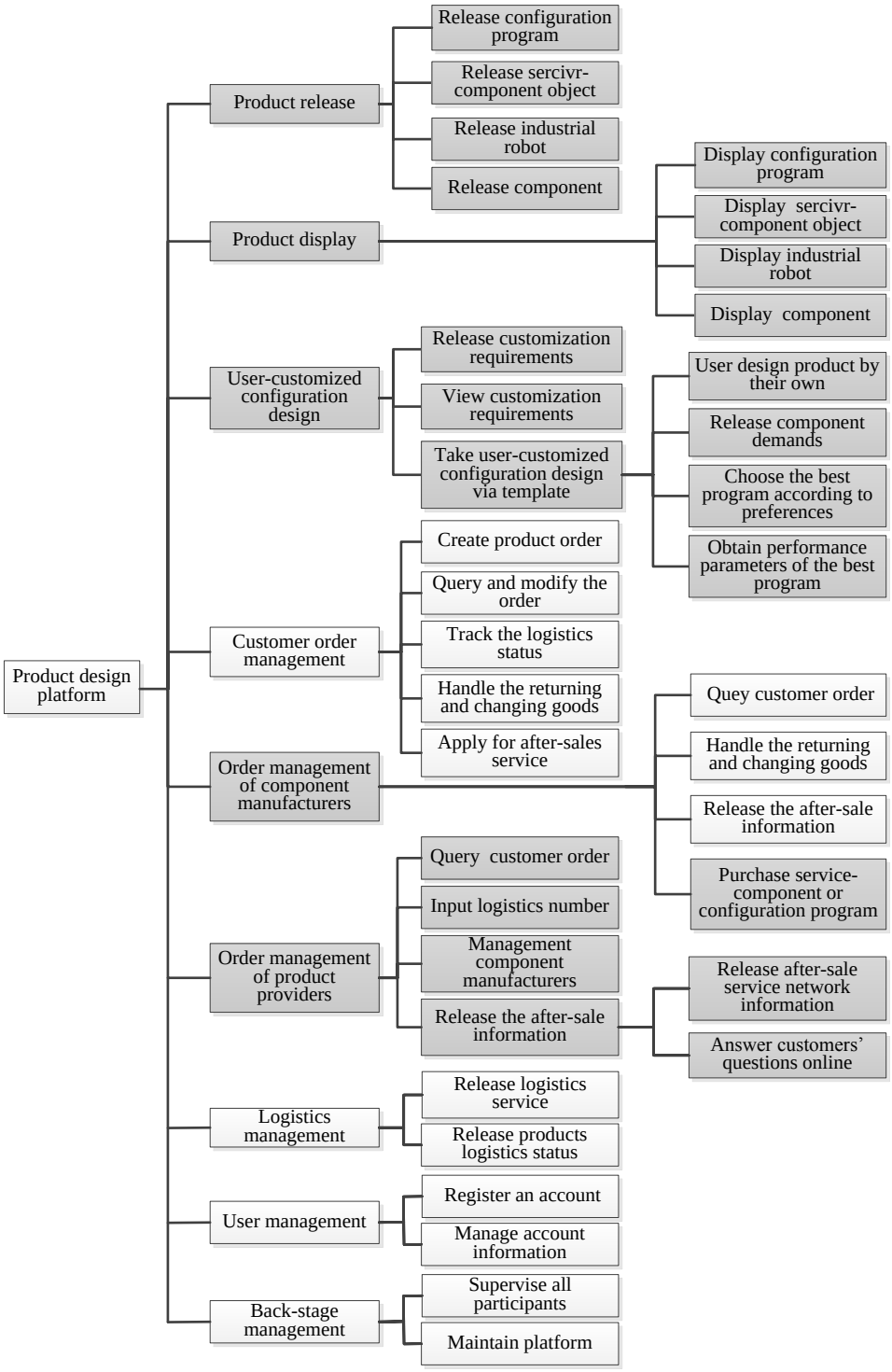


Figure 6-7 Product platform functional block diagram

c) User-customized configuration design. This section is provided for all users with design capacity. Users can release customization requirements here, and these requirements can be seen by other designers who have the ability to solve. Users can also view other customization requirements to see if they have the ability to solve this problem. Finally, users can take the user-customized configuration design on the platform. The final design result can

be used for their own solutions, and also can be released to the platform through the product release module, so that others can share the design results.

d) Order management for component manufacturers. This section includes the sales order management manufacturer of components uploaded by component manufacturers, and also includes the purchase of service-components and configuration programs on the platform. The service-components and configuration programs are design results of designers, and they haven't been manufactured to real components and industrial robots. Component manufacturers can purchase service-components and configuration programs suiting for their own production conditions from the platform, then component manufacturers produce them to real components and industrial robots. After that, component manufacturers release these components and industrial robots to the platform through the product release module.

e) Order management for product providers. Product providers are a bridge between the customers and platform. They are responsible for achieving the products designed by customers. On the platform, the product providers can query the customer's order to know the order logistics information, can entry the logistics number of product which has been assembled for customer inquiries, can publish the evaluation information components, can take rating management for component manufacturers, can publish after-sale information of the product which is assembled by product providers, can publish the after-sale service network information to the platform for customer inquiries, also can answer customers' questions online.

6.2.2 Platform physical configuration design

(1) Select development software. Fireworks and Dreamweaver are used for foreground web design. Both of them are visual development tool. Images can be directly imported into Dreamweaver for web design after being cut in Fireworks. Eclipse is the backend development tool. Development language id Java. Spring MVC framework is selected as Web frameworks. Spring MVC framework is a request driven-based architecture. The running process of the architecture to handle the request is shown in Figure 6-8. When the server receives a user request, the front controller will delegate the task to the controller. The controller creates the appropriate model depending on the task, and a model is returned to controller. The front controller will select the appropriate view to render model data based on the returned model. Then the view response is returned to the user, a request processing is completed. This mode achieves a good separation decoupling of the foreground interface layer and the background data model layer. The interaction and synchronization of foreground

and background is achieved by the middle controller, which makes the development process more simple and flexible.

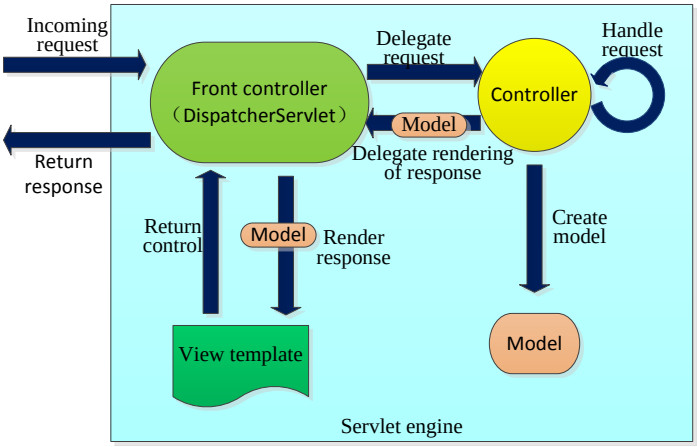


Figure 6-8 The running process of Spring MVC [167]

(2) Select database. Oracle 9i is selected as the platform database. Oracle 9i has great advantages in processing large amounts of data, storing multimedia data such as sound and animation, security, stability. It is very suitable for the design platform development of this paper.

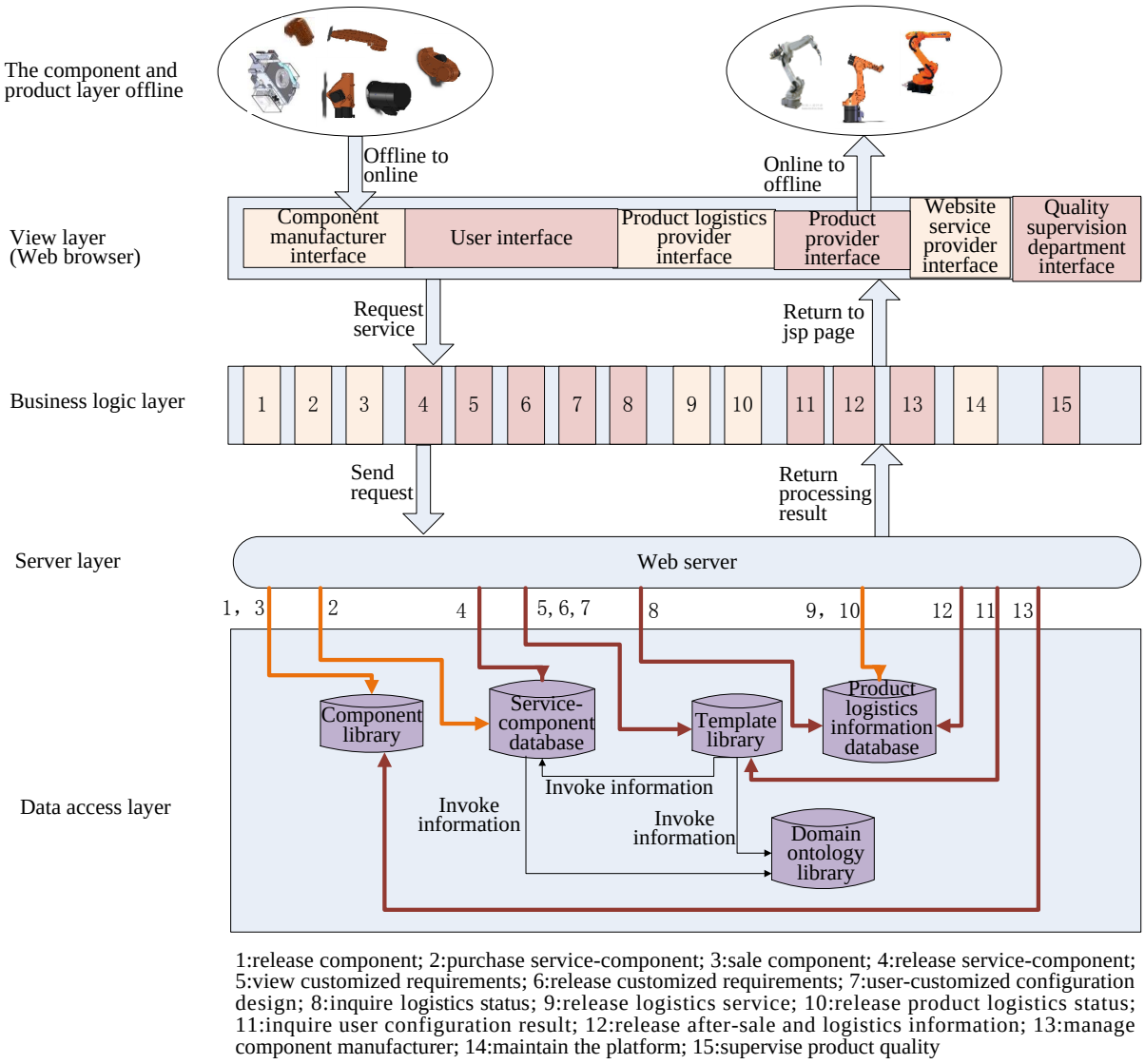
(3) Select computer hardware. Current mainstream computer configuration can meet the design and development work of this paper. Core quad-core CPU, 4G memory, 500G hard disk, Windows 7 computer operating system are selected for the development work.

6.2.3 The platform architecture

Product design platform connects user-customized configuration design online with components and industrial robots offline. The platform architecture is shown in Figure 6-9.

The whole architecture is divided into view layer, business logic layer, server layer, data access layer. View layer and business logic layer are the part that platform users can see. Product designers release the designed service-component to service-component database. Component manufacturers extract design information from the service-component database, and then release the component production information such as type, size, origin, price, etc. to the component library. Users invoke the information in service-component database and domain ontology library through configuration template to design product and create orders. Users also can release new requirements to the template library for product designer to study. Product providers read the order information, assemble components to product offline and release after-sale information on the platform. Product logistics providers store logistics information into the product logistics information database for user query. Quality supervision

departments interact with all the participants through the platform, supervise products and service quality of each participant and ensure that the entire process and the final products meet national standards. Website service service providers provide background maintenance, and also interact with all the participants through the platform to resolve technical problems encountered in using the platform.



1:release component; 2:purchase service-component; 3:sale component; 4:release service-component; 5:view customized requirements; 6:release customized requirements; 7:user-customized configuration design; 8:inquire logistics status; 9:release logistics service; 10:release product logistics status; 11:inquire user configuration result; 12:release after-sale and logistics information; 13:manage component manufacturer; 14:maintain the platform; 15:supervise product quality

Figure 6-9 The platform architecture diagram

6.3 Platform page design

6.3.1 Platform home page

The content of platform home page is shown in Figure 6-10. It is divided into three parts from top to bottom, the top is a title bar which owned by all platform pages, including functions of login, register, function navigation. The left side of middle part is the display bar for service-components and products. More contents will be shown after clicking on the

"More", and then advanced screening can be taken. The details can be seen in section 6.3.2, The right side of middle part shows a list of the most important function for the platform, including user-customized configuration design, release service-component object, release configuration program, release component, release industrial robot, inquire order. Users can access the appropriate interface by clicking button quickly and easily. The function of user-customized configuration design is the most competitive function for the platform, so it is distinguished to other functions with different colors. The contents of bottom are separately the recent news, the platform advantages, advertising position, blogroll. The page of the platform advantages introduces the open platform advantages, including providing full customization function, building web development and manufacturing integrated platform of industrial robots to support users, component manufacturers, designers and all other relevant actors to use, providing more professional electronic catalog to establish industrial robots industry standards.

Figure 6-10 The content of platform home page

6.3.2 Product releasing and display

Different roles can release different contents on platform, and users need to log in before releasing. In the service-component object releasing page, the releaser select the service-component object type, such as actuator, motor, base, etc. Depending on the type, the content of *Atr* on the page is different. For example, the *Atr* elements of motor contain power,

velocity, the moment of inertia, etc.. The *Atr* elements of reducer contain the reduction ratio, the maximum torque, the hardness of tooth surface, installation forms and so on. Object name and *Ob* value is filled by the releaser. The value of *Ma* and *On* with gray color is filled automatically by the platform. Here the instance information is empty, because this information is filled by component manufacturers, not by platform. If the object type is reducer, its page displays as shown in Figure 6-11. Component releasing is similar to service-component releasing. The only difference is that component releasing need to fill the instance information to *Ma*.

Figure 6-11 Service-component object releasing page

The configuration program releasing page is shown in Figure 6-12. The configuration program obtained through user-customized configuration design is released to the platform on this page. The value of *Ob* is the performance parameter value obtained by configuration template, which is written by the releaser. The value of *Ma* and *On* is filled automatically by the platform. The industrial robot releasing is similar to configuration program releasing. Also the only difference is that industrial robot releasing need to fill the instance information to *Ma*.

Figure 6-12 Configuration program releasing page

By clicking the button "More" in home page, more detail contents are shown. For example, click the button "More" of the "service-component object display" in home page, the pop-up page is shown in Figure 6-13. Configuration program releasing page, component releasing page and industrial robot releasing page are all similar to this. This paper won't explain them all here.

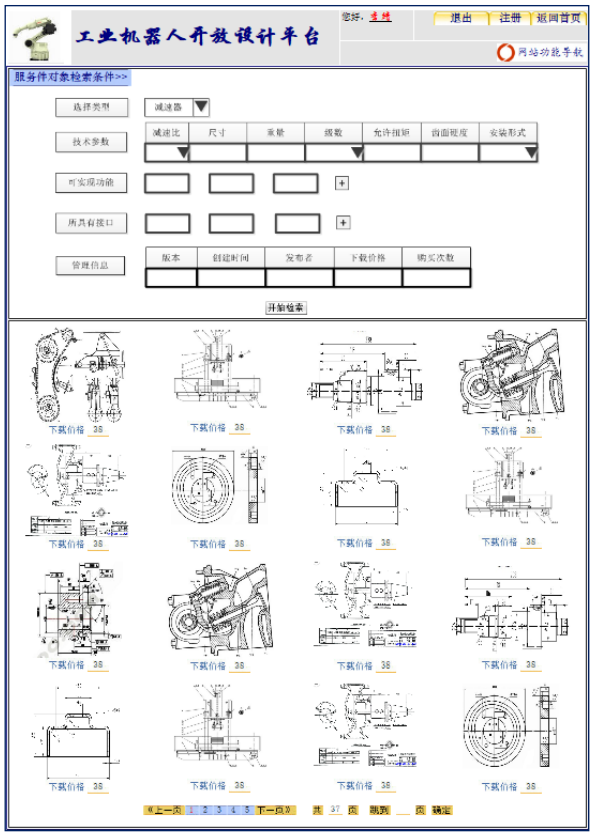


Figure 6-13 Service-component object display page

6.3.3 User-customized configuration design page

Click the button "user-customized configuration design" in home page, and then the page is shown in Figure 6-14. Users can view the released requirements of other people, choose the requirement that they have the ability to solve, release their own requirements to the platform. Users can also click the bottom button "enter user-customized configuration design page", then enter the page which is shown in Figure 6-15.

Figure 6-14 Release/view requirements page

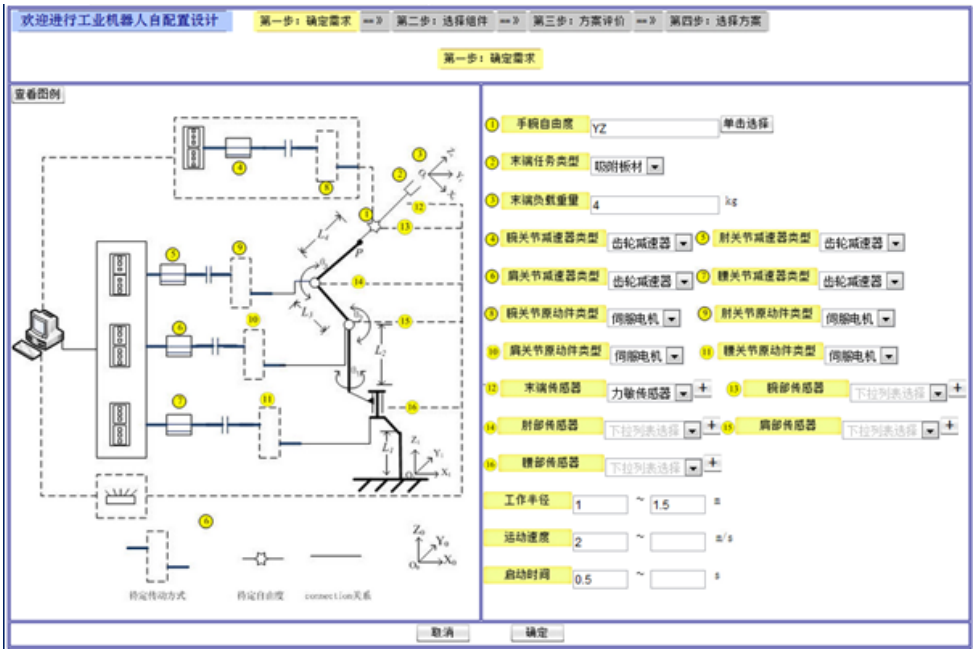


Figure 6-15 The home page for user-customized configuration design

The left of figure 6-15 is the industrial robot template represented by mechanical sketch. Click on the top left of the "Show Legend", the content of table 4-1 in Chapter 4 is displayed, which explains the mechanical sketch elements of configuration template. On the right side, users can enter various parameters to the template according to their needs. When click the button "Click to select" beside the DOF of wrist, the content of table 5-1 in Chapter 5 is shown for users to choose. After all parameters are input, the system will get the industrial robot preliminary structure as shown in Figure 6-16. In the figure, all the component types of service-components have been determined.

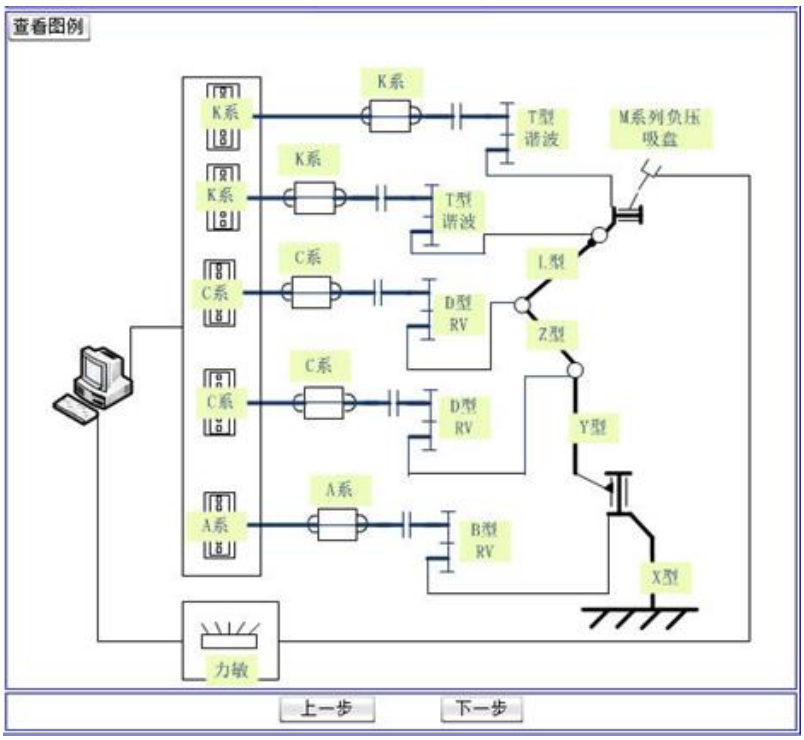


Figure 6-16 Configuration template with determined structure

After the requirement input is completed, the users click on the button "Next step" to enter the page of component selection as shown in Figure 6-17. Users sort component parameters according to their needs, which is shown in Figure 6-18. The brand value, cost performance here is given by the product provider based on past experience. After sorting is completed, the system takes a series of operations including retrieving similar and relevant sercive-components, removing the sercive-components whose value range doesn’t meet the constraints, determining the instantiation order of sercive-components, instantiating the s sercive-components and screening the components. The final configuration program is shown in figure 6-19.

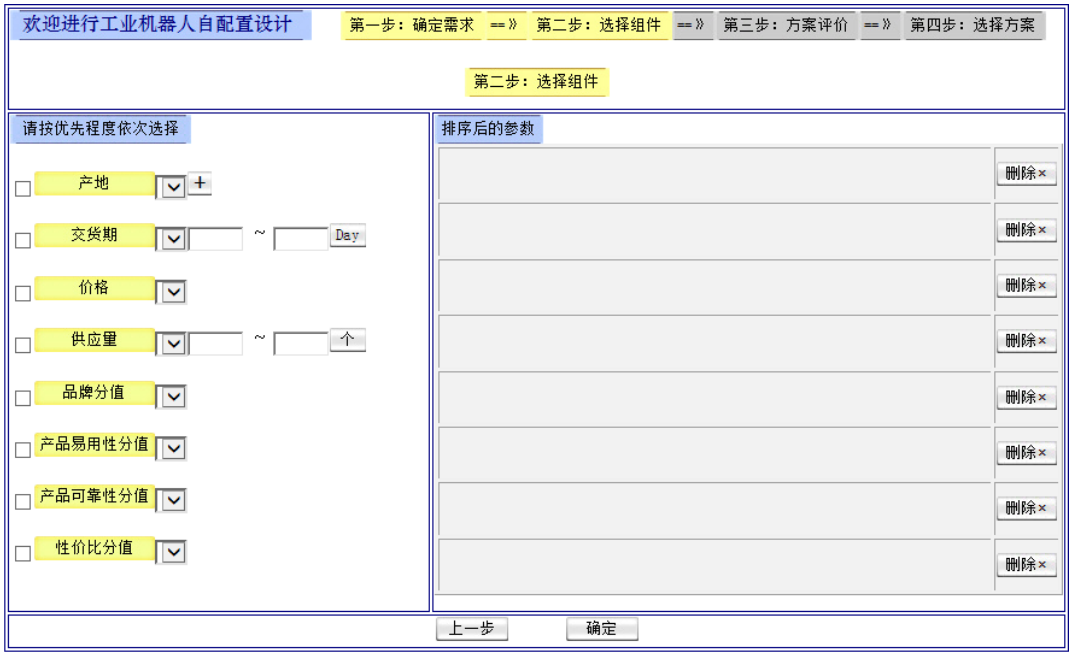
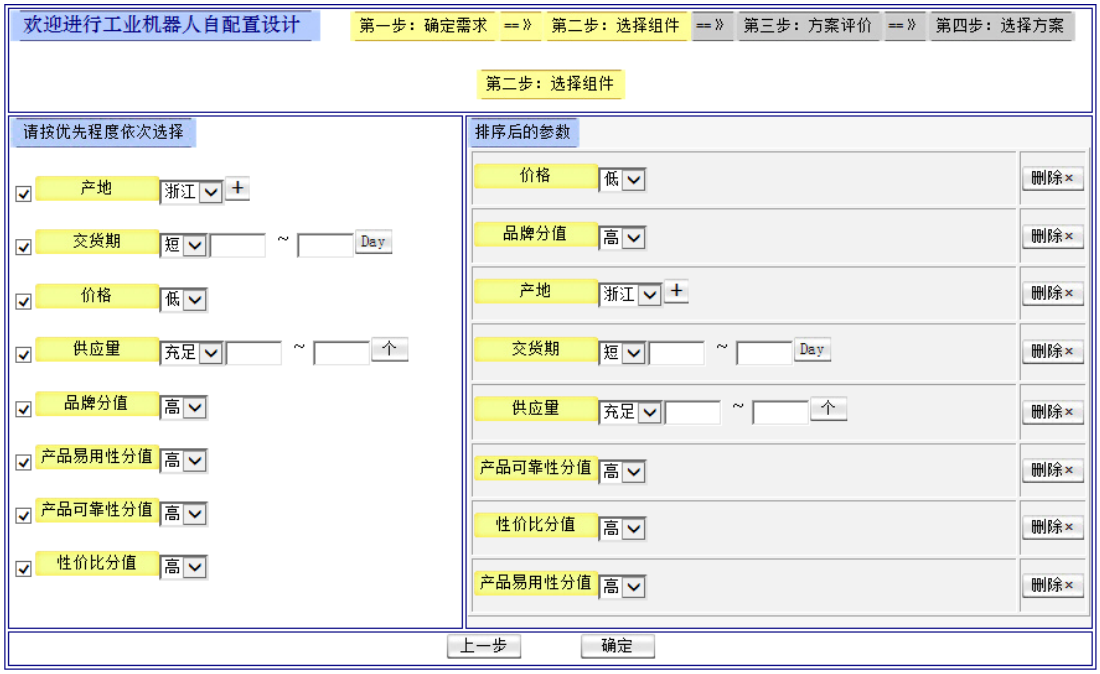


Figure 6-17 The page of determining component filter conditions



Fingure 6-18 Component parameters priority ordering page

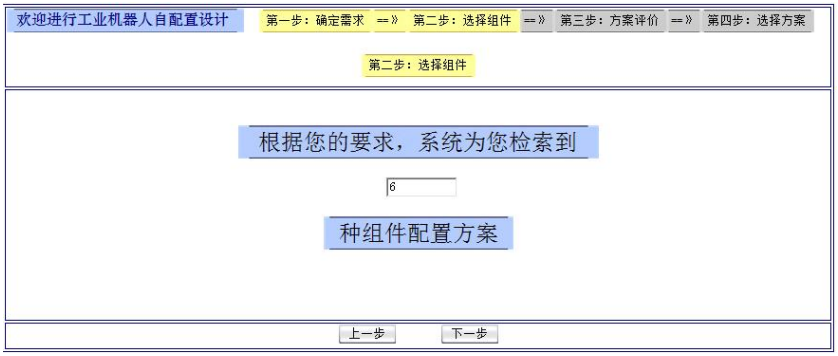


Figure 6-19 Configuration result page

Click on the button "Next step" to enter the configuration program evaluation page, as shown in Figure 6-20. The user enters the value of each performance indicator based on the importance of each performance indicator in his mind. After entering, the user click on the button "OK" to enter the page shown in Figure 6-21. In this page, the user enter the indicator type according to their requirements. When indicator type is interval type, user also needs to enter the maximum value and minimum value.When the input is completed, user clicks on the button "OK", then the system starts to score the six configuration programs based on user's input.

Figure 6-20 Home page of configuration program evaluation

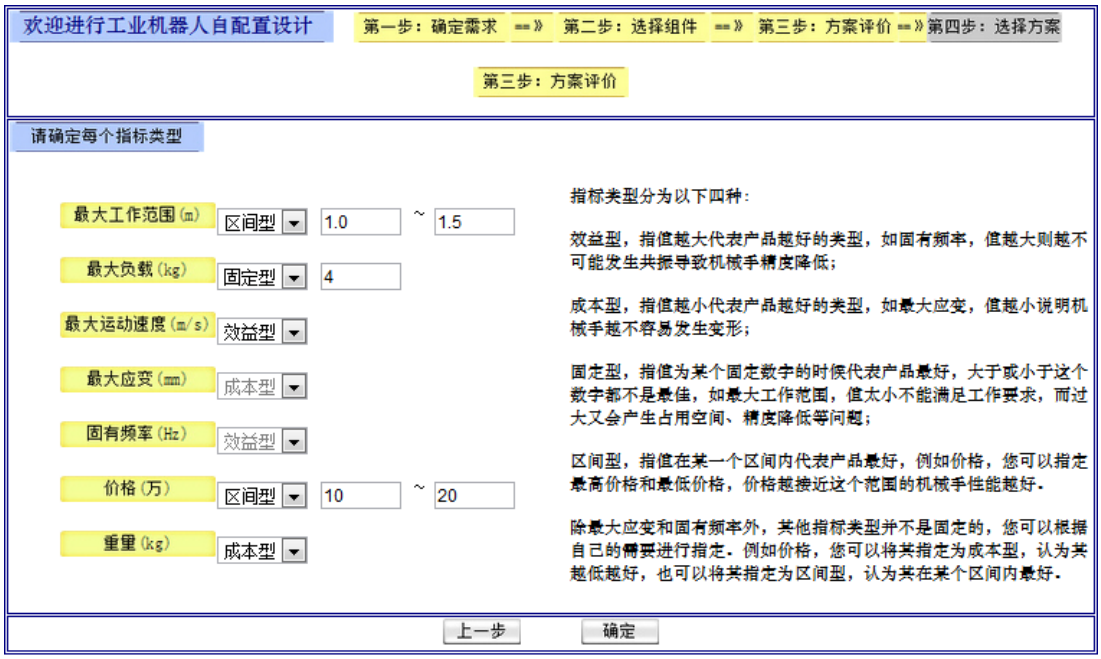


Figure 6-21 The page of determining the performance indicators' type

After system scoring, the page shown as the figure 6-22 is popped up. The figure shows the final score and rank of the programs. The user select the program 6 with the highest score as the final design result. After clicking on the button "OK", the final industrial robot is shown in Figure 6-23. This page shows the three-dimensional map of industrial robot, and also can take the following operations: view the performance indicator value of the program, view the components list of the program, pruchase the product directly by creating order, release the program to the platform for others to view or download. By clicking on the button "View Performance", a page is popped up shown in Figure 6-24. This page shows the indicators' values such as the maximum velocity, the maximum load, price, weight, the workspace graphics simulated by the Simscape model, the deformation cloud graphics, the natural frequencies and the vibration mode graphics simulated by ANSYS model.

Figure 6-22 The page of configuration programs selection

Figure 6-23 User-cuatomized configuretion design result display page

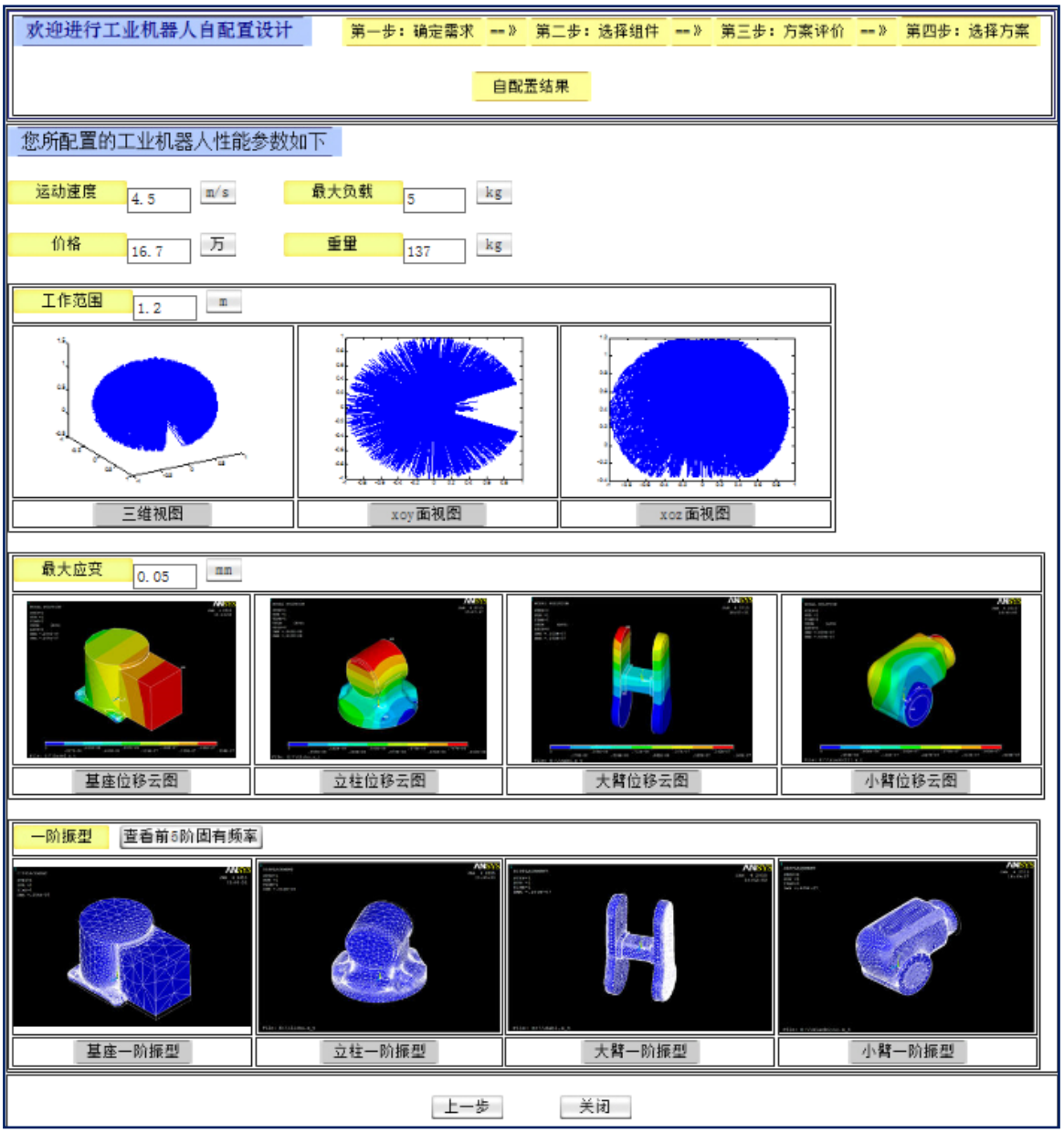


Figure 6-24 Product performance page

6.4 Chapter summary

On the basis of the previous section, this chapter researchs the prototype system design of the open design platform. Firstly, system requirements analysis is carried out, including analyzing the required functions through the use case diagram, taking role-based business process analysis; secondly, then the system design is carried out, including designing platform functional modules based on system requirements analysis, designing physical configuration program of the platform, and drawing the whole architecture of the platform; finally, the main pages of the platform are designed, and the key functions are introduced.

CHAPTER 7 SUMMARY AND OUTLOOK

7.1 Research summary

Industrial robot is a type of mechatronic products containing multiple fields' components such as machinery, electronics, control unit, and computer. It is characterisedby complex structure, long lead times, and high manufacturing costs. However, an industrial robot is required to achieve high degree of customisation for finishing special work. These factors constrained and limited the rapid development of industrial robot.

To solve this problem, thesis work is around the "key technology for industrial robot user-customized configuration design". The thesis studies the key issues in depth including user-customized configuration design pattern, industrial robot components representation, user-customized configuration design template, user-customized configuration design internal algorithm. Based on these, a prototype system is designed. The main work of this thesis includes the following:

(1) Determine research direction by studying the status analysis

The research status of design method such as modular design, product family design, configuration design, reconfigurable robot design, object-oriented design of software, component-based design of software, and the ontology-based knowledge representation is summarized. By analyzing the deficiencies of existing research and future research trends, the research direction is determined combining with the current problems faced by industrial robot design.

(2) Put forward the user-customized configuration design pattern in view of the characteristics of industrial robot

Firstly the similarity of software product development process and hardware product development process at different times is compared, and then point put that the user-customized configuration design pattern drew on service-oriented architecture (SOA) of software development. Secondly the related concepts are expounded, including pattern, configured product, component, service-component, user, user-customized configuration design, design platform, etc. Thirdly, the corresponding business mode of user-customized configuration design pattern is given according to business mode elements, and the key technologies to realize business mode is studied. Through this business mode, more companied can participate in the design and manufacture engineering, and the design iteration is reduced which promote the rapid development of industrial robot industry. Finally, system

dynamics models were established for user-customized configuration design business mode and for traditional business mode of industrial robots by Anylogic simulation software. The simulation analysis proved that the proposed design pattern has obvious advantages in cost and user order response than traditional mode.

(3) Formalized describe the multi-field components in the real world composing an industrial robot

The component-based theory and method are studied, including the formal description of things, ontology representation, componentization and servitization. On this basis, the componentization description model is established for the product parts. And the model is represented as service-component. The service-component includes ontology layer, object layer, business layer. The object instantiation and component instantiation of the service-component can support the users design process. Next, the formation process and extension method of service-component are introduced. Finally, an example of industrial robot components modeling is analyzed. Firstly industrial robot domain ontology is established by protégé, and then the description method, instantiation process and extension method of service-component are illustrated.

(4) Build the user-customized configuration design template for industrial robot

In this chapter, firstly the industrial robot user-customized configuration design template is constructed through analyzing the industrial robot composition structure, and users can obtain the industrial robot meeting constraints through parameters setting. The template contains the internal constraints between components. By calculating the internal constraints, the template can convert parameters input by user to the parameter constraints of service-component. Secondly the kinematics and dynamics analysis on template is taken by Simscape model, and the dynamic parameters such as joint torque, velocity and so on are simulated and output. Thirdly the finite element analysis on template is taken by ANSYS, including statics analysis and modal analysis. Fourthly the parameters flow process in template is analyzed. Finally taking industrial robot user-customized configuration design using configuration template as an example, the configuration template is analyzed in application. The result shows that the user-customized configuration design template can provide users with a good view of the interface, and users can get the final product performance parameters by simple parameter input.

(5) Provide the internal algorithm for user-customized configuration design

Firstly constructs platform-based and user-leading user-customized configuration design

process. The process gives the operations of user, the tasks finished by platform, the model and database needed by platform. Then the internal algorithm to keep the design running smoothly is studied, including the degree of freedom determination, fuzzy demand calculation, and service-component configuration. The service-component configuration contains service-component retrieval based on similarity and relevancy calculation by *On*, service-component selection and instantiation by *Ob*, and service-component object selection by *Ma*. The simple linearity weighted multi-attribute decision making method is used to evaluate the configuration program, then the product that best meeting the user's preference is obtained. Finally a case analysis is taken for the internal algorithm

(6) Design a prototype system of product design platform on the basic of the theory study

Firstly, system requirements analysis is carried out, including analyzing the required functions through the use case diagram, taking role-based business process analysis; secondly, based on system requirements analysis, the system design is carried out. The main work contains two aspects, one is to design platform functional modules such as releasing component and product, releasing service-component, designing product on template, and the other one is to design physical configuration program of the platform and draw the whole architecture of the platform. Finally, the main pages of the platform are designed, and the key functions are introduced.

7.2 The main innovation

The main innovation of this thesis includes the following:

(1) Put forward the user-customized configuration design pattern, and give the activities of the various participants in this pattern. The core of this pattern is to finish the industrial robot design by users themselves. This pattern achieves the user to fully customize on the premise of mass production of industrial robot component. (Chapter 2)

(2) Put forward the concept of service-component based on object-based thinking and ontology. The service-component instantiation contains component instantiation and object instantiation. The instantiation results are components in real world and object abstract thinking respectively. As an expression standard for multi-fields components of industrial robot, the service-component is not only the basic of design process, but also the release standard for components uploaded to the platform. The service-component ensures the openness of the platform and the flexibility of the user-customized configuration design. (Chapter 3)

(3) Establish the user-customized configuration design template. The template can convert user requirements into constraints of service-component. These constraints are the service-component retrieval basis from the database. Simultaneously the final configuration result, i.e. industrial robot, the performance parameters obtained by simulation can be shown with visual form through the template. (Chapter 4,5,6)

(4) Provide the internal algorithm for user-customized configuration design. According to the design process of processing user requirements, configuring service-component, and evaluating configuration program, the internal algorithm ensuring the user-customized configuration design process is studied. Under the guarantee of the algorithm, the user can quickly obtain the product meeting requirements. (Chapter 5)

7.3 Work prospects in future

As the theory and practice of configuration design evolving, user participation in the design is an inevitable trend. Industrial robots are already generally high degree of modularity, and how to enable users to take advantage of these modules to quickly design products meeting their needs is a serious problem. To solve this problem, the thesis studied the possibility and the method of user participation in industrial machine design. Limited time and ability, there are still many problems to be addressed, including:

(1) The thesis only research the industrial robots hardware configuration, but the software control system is also an important aspect for the industrial robot performance. Only with the control system industrial robot can truly moving. Today configuration software is mature gradually, how to configure the industrial robot control system is the next problem to be studied.

(2) The template of this thesis needs to be connected to Simscape and ANSYS to parameter calculation and simulation. But now the calculation and simulation is taken in Simscape and ANSYS and then the parameters are input to template manually. The next step is take secondary development on Simscape and ANSYS, so the data between platform and software can be transmitted automatically.

(3) The established ontology for the industrial robot field is just a primary ontology, and it needs to be revised and improved continuously based on the actual situation, so as to continuously optimize the retrieval efficiency, recall ratio and precision ratio of the service-component.

(4) The functions of the finally prototype system such as creating order, product rating

and so on need to be further improved.

REFERENCES

[1] ZHANG Ning, WANG Tian-mei, XU Hai-xi, LIU Xiao-zheng. Research on E-business Model [J]. Journal of Central University of Finance and Economics, 2004, (2): 68-70.[Chinese]

[2] WANG Zhao-yi. Comparative Study on O2O Model and B2C Model [J]. Journal of Anhui Business College, 2013, 12(3): 28-31.[Chinese]

[3] Harish Patil; Brig. Rajiv Divekar. Inventory Management Challenges For B2C E-Commerce Retailers[J]. Procedia Economics and Finance. 2014,(11):561-571

[4] Wenming Zuo, Qiuping Huang, Chang Fan, Zhenpeng Zhang. Quality management of B2C e-commerce service based on human factors engineering[J]. Electronic Commerce Research and Applications. 2013, (12): 309-320.

[5] Weijia You, Lu Liua, Mu Xia, Chenggong Lv. Reputation inflation detection in a Chinese C2C market[J]. Electronic Commerce Research and Applications. 2011, (10):510-519.

[6] Rajshekhar G. Javalgi, Kenneth D. Hall, S. Tamer Cavusgil. Corporate entrepreneurship, customer-oriented selling, absorptive capacity, and international sales performance in the international B2B setting: Conceptual framework and research propositions[J]. International Business Review. 2014, (23):1193-1202.

[7] Wen Guang Qu, Alain Pinsonneault, Daniel Tomiuk, etc. The impacts of social trust on open and closed B2B e-commerce:A Europe-based study[J]. Information & Management 2014.

[8] Amjad Hadjikhani, Peter LaPlaca. Development of B2B marketing theory[J]. Industrial Marketing Management. 2013, (42):294-305.

[9] Sheena Leek, George Christodoulides. A literature review and future agenda for B2B branding: Challenges of branding in a B2B context[J]. Industrial Marketing Management. 2011, (40): 830-837.

[10] Kin Bee Tay, John Chelliah. Disintermediation of traditional chemical intermediary roles in the Electronic Business-to-Business (e-B2B) exchange world[J]. Journal of Strategic Information Systems. 2011, (20):217-231.

[11] Zhang Bo. O2O：Business Revolution in mobile Internet era [M]. Bei Jing: Machinery Industry Press, 2013. [Chinese]

[12] Dhruv Grewal, Ramkumar Janakiraman, Kirthi Kalyanam，etc. Strategic Online and Offline Retail Pricing: A Review and Research Agenda. Journal of Interactive Marketing, 2010, (24):138-154.

[13] Daniele Scarpi, Gabriele Pizzi, Marco Visentin. Shopping for fun or shopping to buy: Is it different online and offline?[J]. Journal of Retailing and Consumer Services. 2014, (21):258-267.

[[14] Lau, Ronald S M. Mass customization: The next industrial revolution [J]. Industrial Management. 1995, 37(5):18-19.

[15] A. Joneja, N.-S. Lee. Automated configuration of parametric feeding tools for mass customization [J]. Computers and Industrial Engineering. 1998, 35:463-469.

[16] J. K. GERSHENSON, G. J. PRASAD, Y. ZHANG. Product modularity: definitions and benefits[J]. Journal of Engineering Design. 2003,14(3):295-313.

[17] Magnus Persson, Par A hlstrom. Managerial issues in modularising complex products [J]. Technovation. 2006, (26):1201-1209.

[18] J. K. GERSHENSON, G. J. PRASAD, Y. ZHANG. Product modularity: measures and design methods [J]. Journal of Engineering Design. 2004, 15(1): 33-51.

[19] Shih-Wen Hsiao, Ya-Chuan Ko, Chi-Hung Lo, Shih-Ho Chen. An ISM, DEI, and ANP based approach for product family development [J]. Advanced Engineering Informatics. 2012, in press.

[20] CHEN Qiang, LIU Zhi-feng, CAI Li-gang, ZHAGN Guo-jun, GU Pei-hua. Planning Method for Product Platform Based on Axiomatic Design [J]. Computer Integrated Manufacturing Systems, 2010, 16(8): 1587-1596.[Chinese]

[21] WANG Ai-min, MENG Ming-chen, HUANG Jin-yuan. Modular Product Family Design Methodology Based on Design Structure Matrix [J]. Computer Integrated Manufacturing Systems, 2003, 9(3): 214-219.[Chinese]

[22] WANG Wen-dan, QIN X ian-sheng,, YAN Xiu-tian., BAI Jing. Visual design structure matrix method for product design modules identification [J]. Computer Integ rated M anufacturing Systems. 2007, 13(12): 2345-2349. [Chinese]

[23] HOU Liang, TANG Ren-zhong, XU Yan-shen. Reviewoftheor REVIEW of Theory, Key Technologies and Its Application of Modular Product Design [J]. Chinese Journal of Mechanical Engineering, 2004, 40(1): 56-61.[Chinese]

[24] GONG Jing-zhong, QIU Jing, LI Gou-xi, LI Wei. Extension approach for product module variant design [J]. Computer Integrated Manufacturing Systems, 2008, 14(7): 1256-1268.[Chinese]

[25] Yasushi Umeda, Shinichi Fukushige, Keita Tonoike, Shinsuke Kondoh. Product modularity for life cycle design [J]. CIRP Annals - Manufacturing Technology. 2008, (57): 13-16.

[26] Chun-Che Huang, Wen-Yau Liang, Horng-Fu Chuang, Zih-Yu Chang. A novel approach to product modularity and product disassembly with the consideration of 3R-abilities [J]. Computers & Industrial Engineering. 2012, (62): 96-107.

[27]CHEN Yan-hui, ZHOU De-jian, FENG Zhen-jun. Min-max partition modularly method of complex product [J]. Computer Integrated Manufacturing Systems, 2012, 18(1): 9-15.[Chinese]

[28] Victor B. Kreng, Tseng-Pin Lee. Modular product design with grouping genetic algorithm-a case study [J]. Computers & Industrial Engineering. 2004, (46):443-460.

[29] Suiran Yu, Qingyan Yang, Jing Tao, Xia Tian, Fengfu Yin. Product modular design incorporating life cycle issues - Group Genetic Algorithm(GGA) based method [J]. Journal of Cleaner Production. 2011, (19):1016-1032.

[30] Hwai-En Tseng, Chien-Chen Chang, Jia-Diann Li. Modular design to support green life-cycle engineering [J]. Expert Systems with Applications. 2008, (34): 2524-2537.

[31] WANG Hai-jun, WANG Ji-jun, SUN Bao-yuan, LIU Zhi-feng, WEI Xiao-peng. Optimization Design of Modularized Product Family Based on Kernel Platform [J]. Computer Integrated Manufacturing Systems, 2005, 11(2): 162-167.[Chinese]

[32] HUANG Hui, LIANG Gong-qian, SUI Hai-yan. Principle Clustering of Product Families for Mass Customization Design [J]. Journal of Industrial Engineering Management, 2008, 22(3): 110-114.[Chinese]

[33] Meyer, M.H., Lehnerd, A.P.. The Power of Product Platforms: Building Value and Cost Leadership[M]. 1997, Simon & Schuster Ltd.

[34] Tarek AlGeddawy, Hoda ElMaraghy. Design methodology for product family platforms, modularity and parts integration [J]. CIRP Journal of Manufacturing Science and Technology. 2012, in press.

[35] ALLEN, K. R. and CARLSON-SKALAK, S.. Defining product architecture during conceptual design[C]. 1998, Proceedings of the 1998 ASME Design Engineering Technical Conference, Atlanta.

[36] Hou Liang, Tang Renzhong, Xu Yanshen. Review of theory, key technologies and ITS application of modular product design [J]. Chinese journal of mechanical engineering. 2004，40(1):56-62.[Chniese]

[37] Zacharias, N.A., Yassine, A.A.. Optimal Platform Investment for Product Family Design [J]. Journal of Intelligent Manufacturing. 2008, 19(2): 131-148.

[38] Chen, S.L., Jiao, R.J., Tseng, M.M.. Evolutionary Product Line Design Balancing Customer Needs and Product Commonality [J]. CIRP Annals - Manufacturing Technology. 2009, 58:123-126.

[39] Fujita, K., Yoshida, H.. Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes [J]. Concurrent Engineering Research and Applications. 2004, 12:105-118.

[40] Zhihuang, D., Scott, M.J.. Effective Product Family Design using Preference Aggregation [J]. Transactions of the ASME Journal of Mechanical Design. 2006, 128:659-667.

[41] Jianxin jiao, Mitchell M. Tseng, Vincent G. Duffy, Fuhua Lin. Product family modeling for customization [J]. Computers industry engineering. 1998,35:495-498.

[42] Huang Hui, Liang Gongqian, Sui Haiyan. Principle Clustering of Product Families for Mass Customization Design [J]. Journal of Industrial Engineering Engineering Management. 2008，22(3):110-115.[Chniese]

[43] Yuan-Ping Luh, Chih-Hsing Chu, Chih-Chin Pan. Data management of green product development with generic modularized product architecture [J]. Computers in Industry. 2010, 61(3):223-234.

[44] Jeffrey B Dahmus, Javier P Gonzalez-Zugasti, Kevin N Otto. Modular product architecture [J]. Design Studies. 2001, 22(5):409-424.

[45] WANG Shi-wei, TAN Jian-rong, ZHANG Shu-you, JI Yang-jian. Product Configuration Based on Generic Bill of Material [J]. Journal Of Computer-Aided Design &Computer Graphics, 2004, 16(5): 655-659.[Chinese]

[46] DAN Bin, FENG Tao. Product Family Model and Its Configuration Based on GBOM [J]. Chinese Journal of Management, 2005, 2(4): 422-425.[Chinese]

[47] Zhu Shangshang, Luo Shijian, He Ji. Product family design DNA for product visual identity [J]. Journal of Zhejiang University (Engineering Science). 2010, 44(4):715-722.[Chniese]

[48] D Souza B, Simpson T W. A genetic algorithm based method for product family design Optimization [J]. Engineering Optimization. 2003, 35(1):1-18.

[49] Kikuo Fujita. Product variety optimization under modular architecture [J]. Computer-Aided Design. 2002, 34(12):953-965.

[50]WANG Kexi, YUAN Ji-jun, CHEN Wei-min, QUAN Chun-guang. A Two-stage Multi-objective Intelligent Optimization Algorithm for Single Platform Based Product Family Design [J]. The Chinese Mechanical Engineering, 2011, 22(17): 2097-2106.[Chinese]

[51] LUO Xing-gang, CAI Li-qing, K WONG C.K. Multi-objective optimization method for product family design [J]. Computer Integrated Manufacturing Systems, 2011, 17(7): 1345-1356.[Chinese]

[52] Wei Wei, Yixiong Feng, Jianrong Tan, Zhongkai Li. Product platform two-stage quality optimization design based on multi objective genetic algorithm [J]. Computers and Mathematics with Applications. 2009, (57):1929-1937.

[53] NIE Yong, YIN Guofu, ZHAO Xiufen. Product Family Design Based on Mechanism of Population Evolution [J]. JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY. 2012, 47(3):526-533. [Chinese]

[54]HUA Er-tian, XIAO Jun-jun, LIU Ke-hong, ZHOU Ke. Method of knowledge reduction for product family design based on fuzzy clustering and rough set [J]. Application Research of Computers.2011, 28(11): 4064-4067. [Chinese]

[55] Bruno Agard, Bernard Penz. Simulated annealing method based on a clustering approach to determine bills of materials for a large product family [J]. International journal of Production Economics. 2009, (117) :389-401.

[56] Tarek AlGeddawy, Hoda ElMaraghy. Reactive design methodology for product family platforms, modularity and parts integration [J]. CIRP Journal of Manufacturing Science and Technology. 2012, in press.

[57] Sanjay Mittal, Felix Frayman. Towards a Generic Model of Configuration Task[Al. International Joint Conference on Artificial Intelligence IJCAI, Detroit，1989，2:1395-1401.

[58] Alessio Trentin, Elisa Perin, Cipriano Forza. Overcoming the customization-responsiveness squeeze by using product configurators: Beyond anecdotal evidence[J]. Computers in Industry. 2011, 62:260-268.

[59] Timo Asikainen, Tomi Männistö, and Timo Soininen. Using a Configurator for Modelling and Configuring Software Product Lines based on Feature Models [C]. In Proceedings of Software Variability Management for Product Derivation - Towards Tool Support at International Workshop of SPLC, (2004).

[60] Lars Hvam, Klaes Ladeby. An approach for the development of visual configuration systems[J]. Computers & Industrial Engineering. 2007, 53:401-419.

[61] Sehyun Myung,Soonhung Han. Knowledge-based parametric design of mechanical products based on configuration design method[J]. Expert Systems with Applications. 2001,21:99-107.

[62] T.W. Carnduff, J.S. Goonetillake. Configuration management in evolutionary engineering design using versioning and integrity constraints[J]. Advances in Engineering Software. 2004, 35:161-177.

[63] Peng Gaoliang , Wang Gongdong, Liu Wenjian, Yu Haiquan. A desktop virtual reality-based interactive modular fixture configuration design system[J]. Computer-Aided Design. 2010,42 :432-444.

[64] Shih-Wen Hsiao, Elim Liu. A structural component-based approach for designing product family [J]. Computers in Industry. 2005, (56):13-28.

[65] Xuan F. Zha, Ram D. Sriram. Platform-based product design and development: A knowledge-intensive support approach [J]. Knowledge-Based Systems. 2006, (19):524-543.

[66] Egon Ostrosi, Alain-Jérôme Fougères. Optimization of product configuration assisted by fuzzy agents [J]. International Journal on Interactive Design and Manufacturing. 2011, (5):29-44.

[67] H.S. Wang, Z.H. Che, M.J. Wang. A three-phase integrated model for product configuration change problems [J]. Expert Systems with Applications. 2009, (36): 5491-5509.

[68]]H-E. Tseng, C-C. Chang, and S-H. Chang. Applying case-based reasoning for product configuration in mass customization environments [J]. Expert Systems with Applications. 2005, 29(4):913-925.

[69]Yan Wei Zhao, F. Zhang, M.Y. Zhang, etc. Extension Case-Based Reasoning for Product Configuration Design [J]. Advanced Materials Research. 2009, 69-70: 616-620.

[70] F. S. Zeng, Y. Jin. Study on product configuration based on product model [J]. International Journal of Advanced Manufacturing Technology. 2007, 33: 766-771.

[71] Gang Hong, Deyi Xue, Yiliu Tu. Rapid identification of the optimal product configuration and its parameters based on customer-centric product modeling for one-of-a-kind production [J]. Computers in Industry. 2010, 61: 270-279.

[72] Anders Haug, Lars Hvam, Niels Henrik Mortensen. A layout technique for class diagrams to be used in product configuration projects [J]. Computers in Industry. 2010, 61:409-418.

[73] Xuan F. Zha, Ram D. Sriram. Platform-based product design and development: A knowledge-intensive support approach [J]. Knowledge-Based Systems. 2006, 19:524-543.

[74] F. Salvadora, C. Forza. Configuring products to address the customization responsiveness squeeze: A survey of management issues and opportunities[J]. International Journal of Production Economics. 2004, 91:273-291.

[75] Egon Ostrosi, Alain-Jérôme Fougères, Michel Ferney. Fuzzy agents for product configuration in collaborative and distributed design process [J]. Applied Soft Computing. 2012, In Press.

[76] Jianxin (Roger) Jiao, Timothy W. Simpson, Zahed Siddique. Product family design and platform-based product development: a state-of-the-art review [J]. Journal of Intelligent Manufacturing. 2007, 18:5-29.

[77] Randall Taylor, Terwiesch Christian, Ulrich Karl T. Principles for User Design of Customized Products [J]. California Management Review. 2005, 47(4):68-85.

[78] Jianxin (Roger) Jiao, Martin G. Helander. Development of an electronic configure-to-order platform for customized product development [J]. Computers in Industry. 2006, 57:231-244.

[79] Zhu Yun, Chen Xinzhao, Zhang Li, Zhang Jianjun. Study on Generalized Product Configuration Model Based on Customer- driven [J]. Chinese Mechanical Engineering. 2005,16 (33):1175-1179.[Chniese]

[80] Zhang Meng, Li Guoxi, Gong Jingzhong, Wu Baozhong. Technology of product configuration design based on ordered-tree[J].Computer Intergrated Manufacturing Systems. 2010,16(11): 2333-2340. .[Chniese]

[81] QIN Yan-hong , DAN Bin , ZHANG Xu-mei. Product Configuration Based on Modular Product Family and Translation of Customer Requirement [J]. Industrial Engineering and Management. 2008, (1):21-26. [Chniese]

[82] LU Chang-ming, ZHANG Li-bin, J I ANG J ian-dong, etc. Rapid configuration design approach based on design templates [J]. Computer Integ rated M anufacturing Systems. 2009, 15(3):425-430. .[Chniese]

[83] A. Faiña, F. Orjales, F. Bellas. First Steps towards a Heterogeneous Modular Robotic Architecture for Intelligent Industrial Operation. ?

[84] M. Pacheco, M. Moghadam, A. Magnusson, etc. Fable: Design of a Modular Robotic Playware Platform. IEEE International Conference on Robotics and Automation (ICRA). Karlsruhe, Germany, May 6-10, 2013.

[85] Schmitz D, Khosla P, Kanade T. The CMU Reconfigurable Modular Manipulator System [C]. Proceeding of the International Symposium and Exposition on Robots. Sydney,1988: 473-488.

[86] Liu Mingrao, Tan Dalong, Li Bin.Status and Development of Reconfiguration Modular Robots [J]. Robot. 2001,23(3):275-279.

[87] WANG Bing, JIANG Zhen. Review on the status and development of modular reconfigurable robot technology [J]. MECHANICAL & ELECTRICAL ENGINEERING MAGAZINE. 2008, 25(5):1-4. [Chniese]

[88] H.Y.K. Lau, A.W.Y. Ko, T.L. Lau. The design of a representation and analysis method for modular self-reconfigurable robots [J]. Robotics and Computer-Integrated Manufacturing. 2008, (24):258-269.

[89] W.J. Zhang, S.N. Liu, Q. Li. Data/knowledge representation of modular robot and its working environment [J]. Robotics and Computer Integrated Manufacturing. 2000, (16):143-159.

[90] FUJITA M, KITANO H, KAGEYAMA K. A Reconfigurable Robot Platform [C]. IEEE Robotics and autonomous Systerns. 1999:119-132.

[91] MATSUMARU T. Design and Control of the Modular Robot System: TOMMS [C]. IEEE Robotics＆Automation. 1995:2125-2131.

[92] Jason Lemay, Leila Notash. Configuration engine for architecture planning of modular parallel robots [J]. Mechanism and Machine Theory. 2004, (39):101-117.

[93] Z. M. Bi and W. J. Zhang. Concurrent Optimal Design of Modular Robotic Configuration [J]. Journal of Robotic Systems. 2001,18(2):77-87.

[94] José Baca, Manuel Ferre, Rafael Aracil. A heterogeneous modular robotic design for fast response to a diversity of tasks [J]. Robotics and Autonomous Systems. 2012, (60): 522-531.

[95] Mehdi Tarkian, Johan Ölvander, Xiaolong Feng. Design Automation of Modular Industrial Robots [C]. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2009). 2009 , San Diego, California, USA.

[96] Hossein Sadjadi, Omid Mohareri, Mohammad AminAl-Jarrah, Khaled Assaleh. Design and implementation of HexBot: A modular self-reconfigurable robotic system [J]. Journal of the Franklin Institute. 2012, 349(7):2281-2293.

[97] S.V. Shah, S.K. Saha, J.K. Dutt. Modular framework for dynamic modeling and analyses of legged robots [J]. Mechanism and Machine Theory. 2012, (49): 234-255.

[98] Z.M. Bi, W.A. Gruverb., W.J. Zhang, S.Y.T. Lang. Automated modeling of modular robotic configurations [J]. Robotics and Autonomous Systems. 2006, (54):1015-1025.

[99] WEI Yan-hui, ZHAO Jie, GAO Yan-bin, etc. A solution method for kinematics of reconfigurable modular robots [J]. JOURNAL OF HARBIN INSTITUTE OF TECHNOLOGY. 2010, 42(1):133-138. [Chniese]

[100]REN Zong-wei, ZHU Yan-he, ZHAO Jie, ZHANG Yu-hua. Motion and Self-reconfiguration Planning of Self-reconfigurable Robot Based on Windmill-like Meta-module [J]. Journal of Harbin Engineering University, 2009, 30(4): 436-440.[Chinese]

[101] Salemi B, Moll M, Shen W M. SUPERBOT: A deployable, multi—functional, and modular se1f reconfigurable robotic system [C]. IEEE, RSJ International Conference on Intelligent Robots and Systems. Piscataway，NJ，USA：IEEE，2006: 3636-3641.

[102]]R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, A.J. Ijspeert. Exploring adaptive locomotion with YaMoR- a novel autonomous modular robot with Bluetooth interface [J]. Journal of Industrial Robot. 2006, 33 (4):285–290.

[103] Nicolae Plitea,DorinLese,DoinaPisla, CalinVaida. Structural design and kinematics of a new parallel reconfigurable robot [J]. Robotics and Computer-Integrated Manufacturing. 2013, (29):219-235.

[104] Andreas Lyder, Ricardo Franco Mendoza Garcia and Kasper Stoy. Mechanical Design of Odin, an Extendable Heterogeneous Deformable Modular Robot. IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France, Sept, 22-26, 2008.

[105]SUN Zhi-jie, WANG Shan-jun, ZHANG Xue-xin. On the Development Status and Tendency of Industrial Robots [J]. Journal of Jilin Teachers Institute of Engineering and Technology, 2011, 27(7): 61-62.[Chinese]

[106]Wang Weiguo. Object oriented method of software engineering[J].Association for science and Technology Forum.2012,(3):76-77.[Chinese]

[107]Tang Shengqun,Tang, Taozhou. Software Architecture and Component-based Software Engineering[J]. Computer Engineering.1998, 24(8): 32-35.[Chinese]

[108] Mikael Akerholm, Jan Carlson, Johan Fredriksson， etc. The SAVE approach to component-based development of vehicular systems [J]. The Journal of Systems and Software. 2007, (80):655-667.

[109]NIU De-li,MEN Bao-hong,XI Qin.Component Technique and Its Application in GIS[J].Journal of Institute of Surveying and Mapping. 2000, 17(4):265-268.

[110] Ivica Crnkovic, Magnus Larsson. Challenges of component-based development [J]. The Journal of Systems and Software. 2002, (61): 201-212.

[111] Thareendhra Wijayasiriwardhane, Richard Lai. Component Point: A system-level size measure for Component-Based Software Systems [J]. The Journal of Systems and Software. 2010, (83):2456-2470.

[112] Patricia López Martínez, Laura Barros, José M. Drake. Design of component-based real-time applications [J]. The Journal of Systems and Software. 2012, in press.

[113] Cengiz Togay, Ali H. Dogru, John Urcun Tanik. Systematic Component-Oriented development with Axiomatic Design [J]. The Journal of Systems and Software. 2008,(81):1803-1815.

[114] Shi-Ming Huang, Chih-Fong Tsai, Po-Chun Huang. Component based software version management based on a Component-Interface Dependency Matrix. The Journal of Systems and Software. 2009, (82):382-399.

[115] Majdi Abdellatief, Abu Bakar Md Sultan, Abdul Azim Abdul Ghani, Marzanah A. Jabar. A mapping study to investigate component-based software system metrics [J]. The Journal of Systems and Software. 2012, in press.

[116] Ismail, S., Wan-Kadir, W., Saman, Y.M., Mohd-Hashim, S.Z.. A review on the component evaluation approaches to support software reuse [C]. In: Proceeding of Information Technology, 2008. ITSim 2008. International Symposium on: IEEE, pp. 1–6.

[117] Kalaimagal, S., Srinivasan, R., A retrospective on software component quality models [J]. SIGSOFT Software Engineering Notes. 2008, 33 (6):1–10.

[118]ANG Ya-luo, WANG Run-xiao, KU Xiang-chen, WANG L-i gang. Development of Configuration Conception[J].Application Research of computers.2006, (9):13-18.[Chinese]

[119] Bai Jing. Research on key issues of modularized configuration design for CNC equipment [D]. 2010, Xi'an: Northwestern Polytechnical University. [Chniese].

[120] Shi Liangrong. Application of Fieldbus and Configure Software in Sewage Treatment Project [J]. Industrial Control Computer. 2006.23(3):90-91. [Chniese].

[121] Jiang Xingjia. Ethernet Configuration Based onWINCC, KINGVIEW and S7-300PLC [J]. Automation applications. 2011.(2):38-39. [Chniese].

[122] Hu Haomin, Wang Zejie. Design and Application of Real-Time Monitoring System Based on Distributed Control [J]. JOURNAL OF SHANGHAI UNIVERSITY OF ENGINEERING SCIENCE. 2010, 24(4): 342-345. [Chniese].

[123] Li Bin, Zhou Yun-fei, Tang Xiao-qi. A research on open CNC system based on architecture/component software reuse technology [J]. Computers in Industry. 2004, (55):73-85.

[124] Jeremy R. Conway, Charlie A. Ernesto, Rida T. Farouki,, Mei Zhang. Performance analysis of cross-coupled controllers for CNC machines based upon precise real-time contour error measurement [J]. International Journal of Machine Tools & Manufacture. 2012, (52):30-39.

[125] M. Minhat, V. Vyatkin,X.Xu, S. Wong, Z. Al-Bayaa. A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks [J]. Robotics and Computer-Integrated Manufacturing. 2009, (25):560– 569.

[126]http://www.gongkong.com/webpage/paper/200809/2008091214341300003.htm.

[127] Lemay J, Notash L. Configuration engine for architecture planning of modular parallel robots. Mechanism and Machine Theory. 2004, 39:101-117.

[128] Bi ZM, Zhang WJ. Concurrent Optimal Design of Modular Robotic Configuration. Journal of Robotic Systems. 2001,18(2):77-87.

[129] Thomas D. Miller, Per Elgård. Defining Modules, Modularity and Modularization-Evolution of the Concept in a Historical Perspective. Design for Integration in Manufacturing, proceedings of the 13th IPS Research Seminar, Fuglsoe 1998.

[130] T. Fukuda, S. Nakagawa, Dynamically reconfigurable robotic system, in: Proc. Conf. IEEE Int. Robotics and Automation, vol. 3, 1988, pp. 1581–1586.

[131] Wurst KH. The conception and construction of a modular robot system. Proceedings of the International Symposium on Indus-trial Robotics, Belgium, 1986, p. 37}44.

[132] Timo Asikainen, Tomi Männistö, and Timo Soininen. Using a Configurator for Modelling and Configuring Software Product Lines based on Feature Models. in Workshop on Software Variability Management for Product Derivation in conjunction with Software Product Line Conference. 2004: Boston, Massachusetts, USA.

[133] D.Pavlic, M.Storga, N.Bojcetic, D.Marjanovic. Generic Product Structure of the Configurable Product[C]. International Design conference-Design 2004, Dubrovnik, May 18-21, 2004.

[134] SoininenT, StumptnerM. Configuration. Artificial Intelligence for Engineering Design. Analysis and Manufacturing, 2003, 17(1):1-2.

[135] YONG M ingpei, YU Xiongqing. Aircraft Family Design Using Modular Product Pplatform Methodology-An Exploratory Study[J]. Aircraft design. 2006, (4):30-36. (Chinese)

[136] Jiao J, Simpson TW, Siddique Z. Product family design and platform-based product development: a state-of-the-art review. Journal of Intelligent Manufacturing. 2007, 18:5-29.

[137] A. Brunete, M. Hernando, E. Gambao, J.E. Torres. A behavior-based control architecture for heterogeneous modular, multi-configurable, chained micro-robots. Robotics and Autonomous Systems 2012, 60:1607–1624.

[138] LauHYK, KoAWY, LauTL. The design of a representation and analysis method for modular self-reconfigurable robots. Robotics and Computer-Integrated Manufacturing. 2008, 24:258-269.

[139] Antuela A. Tako, Stewart Robinson. The application of discrete event simulation and system dynamics in the logistics and supply chain context [J]. Decision Support Systems. 2012, (52) :802-815.

[140] Guohua C, Genbao Z, XuelinL.Study on dynamics behavior of supply chain with various operational modes. Computer Integrated Manufacturing System. 2011, 17(1):133-141. (Chinese)

[141] BAI Jing,QIN Xian-sheng,ZHANG Shun-qi,TAN Xiao-qun. Product configuration design based on module extension [J]. Computer Integr ated Manufacturing Sy stems, 2009, (11):2089-2095.(Chinese)

[142] CHEN Lei,PAN Xiang,YE Xiu-zi,.etc. Ontology-based product knowledge representation and retrieval [J]. Journal o f Zhejiang Univ ersity(Engineering Science), 2008, (12):2037-2042. (Chinese)

[143] Wu Peng, Wang Yuefen, Ding Shengchun, etc. Mechanical Product Design Knowledge Representation Based on Ontology [J]. Intelligence Theory and Application, 2013, (10):91-95. (Chinese)

[144] Michele Ceccarelli, Luigi Cerulo, Antonella Santone. De novo reconstruction of gene regulatory networks from time series data, an approach based on formal methods[J]. Methods. 2014, (69):298-305.

[145] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition. 1993,5(2):199-220.

[146] Guarino N. Formal Ontology and Information Systems. In Guarino N (Eds.), Formal Ontology in Information Systems, Proceedings of FOIS’98, Trento, Italy, 1998,3-15.

[147] Perez A G，Benjamins V R. Overview of KnowIedge Sharing and Reuse Components：Ontotogies and ProbIem-Solving Methods. In： StockhoIm V R， Benjamins B，

Chandrasekaran A， eds. Proceedings of the IJCAI-99 workshop on OntoIogies and ProbIem-SoIving Methods（KRR5）1999,1-15.

[148] WANG Hong-wei,DUAN Yong-rui,JIANG Fu. Visualization of Ontology Model Based on UML Extension [J]. Journal of Industrial Engineering Engineering Management, 2006, (3):67-73. (Chinese)

[149] Noy N. F., McGuinness D. L., Ontology development 101: a guide to creating your first ontology. Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880: 2001.

[150] Deng Zhihong, Tang Shiwei, Yang Dongqing, Zhang Ming. An Intra-Algebra in Ontology[J]. Computer Engineering and Application, 2001, (23):7-8. (Chinese)

[151] Guoliang Yang. Industrial robot dynamics simulation and finite element analysis [D]。Wuhan: Huazhong University of Science and Technology, 2007. (Chinese)

[152] Lars Hvam, Klaes Ladeby. An approach for the development of visual configuration systems[J]. Computers & Industrial Engineering. 2007, 53:401-419.

[153] Sehyun Myung,Soonhung Han. Knowledge-based parametric design of mechanical products based on configuration design method[J]. Expert Systems with Applications. 2001,21:99-107.

[154] Wei Wei, Yixiong Feng, Jianrong Tan, Zhongkai Li. Product platform two-stage quality optimization design based on multiobjective genetic algorithm[J]. Computers and Mathematics with Applications. 2009, 57:1929-1937.

[155] http://gongjushu.cnki.net/crfdhtml/r200705041/r200705041.6e0c4d.html.

[156] Rahul Chougule, Vineet R. Khare, Kallappa Pattada. A fuzzy logic based approach for modeling quality and reliability related customer satisfaction in the automotive domain. Expert Systems with Applications. 2013, (40):800-810.

[157] Konstantina Kamvysi, Katerina Gotzamani, Andreas Andronikidis, Andreas C. Georgiou. Capturing and prioritizing students' requirements for course design by embedding Fuzzy-AHP and linear programming in QFD. European Journal of Operational Research. 2014, (237):1083-1094.

[158] Yuanyuan Liu, Jian Zhou, Yizeng Chen. Using fuzzy non-linear regression to identify the degree of compensation among customer requirements in QFD. Neurocomputing. 2014,(142):115-124.

[159] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. Proceedings of IJCAI(International Joint Conference on Artificial Intelligence). Canada, August, 1995.(1):448-453.

[160] Yong Jiang, Xinmin Wang, Hai-Tao Zheng. A semantic similarity measure based on information distance for ontology alignment. Information Sciences,2014. (278): 76-87.

[161] Parisa D. Hossein Zadeh, Marek Z. Reformat. Assessment of semantic similarity of concepts defined in ontology. Information Sciences,2013.250:21-39.

[162] David Sánchez, Montserrat Batet. A semantic similarity method based on information content exploiting multiple ontologies. Expert Systems with Applications,2013. (40):1393-1399.

[163] David Sánchez, Montserrat Batet, David Isern, Aida Valls. Ontology-based semantic similarity: A new feature-based approach. Expert Systems with Applications,2012. (39): 7718-7728.

[164] Albert Solé-Ribalta, David Sánchez, Montserrat Batet, Francesc Serratosa, Towards the estimation of feature-based semantic similarity using multiple ontologies. Knowledge-Based Systems, 2013.Volume 55:101-113.

[165] SabouM, RichardsD, Van SplunterS. An Experience Report on Using DAML-S. The Twelfth International World Wide Web Conference Workshop on E-Services and the Semantic Web,2003.

[166] Xu Zeshui. Uncertain Multiple Attribute Decision Methods and Applications [M]. Beijing: Tsinghua University Press. 2004:8. (Chinese)

[167] http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html.